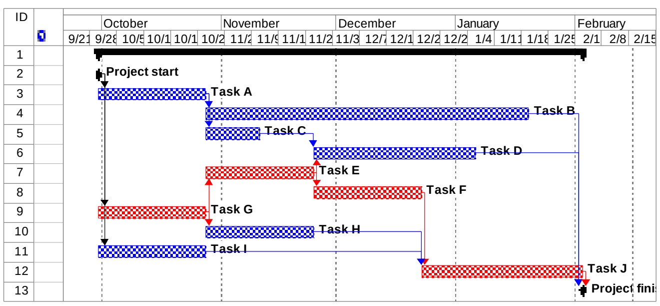
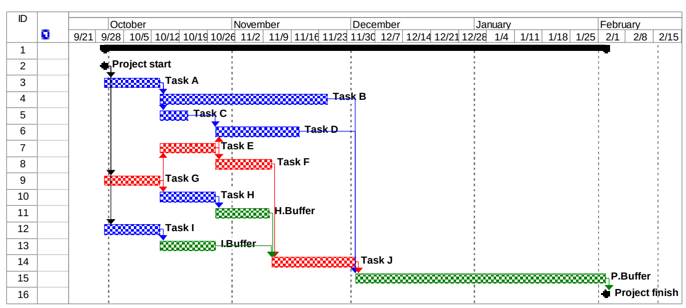
University of Twente

School of Management and Governance

Critical Chain Project Management (CCPM) at Bosch Security Systems (CCTV) Eindhoven

A Survey to explore improvement opportunities in the scheduling and monitoring of product development projects





Master of Science Thesis

Farhad Dilmaghani

August 2008

Critical Chain Project Management (CCPM) at Bosch Security Systems (CCTV) Eindhoven

A survey to explore improvement opportunities in the scheduling and monitoring of product development projects

Farhad Dilmaghani

August 2008

University of Twente
Enschede - The Netherlands

School of Management and Governance

University Supervisors:

Andreas Hartmann
Marco Schutten

BOSCH

Bosch Security Systems Eindhoven

Company Supervisor:

Rob Jansen

Acknowledgements

This research assignment is the final part of my course Industrial Engineering and Management at the Twente University.

I would like to acknowledge all the people that provided support during this graduation project.

Firstly, I would like to expresses my appreciation for the Bosch Security System Eindhoven Co. and in particular to express my sincere gratitude to Rob Jansen my company supervisor for his continuous encouragement and support during the course of this work. I would like also to thank Paul Merkus, Ted Groningen, Jo Dirks, Eric van de Leur and those individuals at the software development department who supported me and took the time to answer to my questions and provided me the necessary information.

Also, I would like to thank my thesis supervisors, Andreas Hartmann and Marco Schutten for their valuable guidance and feedback in the course of this project.

Finally I would like to thank my family and friends who encouraged me during this work.

Farhad Dilmaghani

August 2008

Management summary

Bosch Security Systems Eindhoven Co. (Bosch CCTV) is engaged in development projects for advanced digital video security systems, mainly cameras and recorders. Because of market competition, these developing products get increasingly more functions which lead to increasing complexity and uncertainty. To cope with this increasingly uncertainty, to improve protection of projects from delays, and to increase schedule performance and reliability, Bosch CCTV decided to embark on the Critical Chain Project Management (CCPM) system at its development division. For product development organizations, meeting of project due dates is the vital element for successful introduction of a new product on the market. However, despite the promising benefits and many successful stories of CCPM deployment, Bosch CCTV did not achieve the main benefits of Critical Chain approach as presumed in the literature e.g., reducing project duration, gaining more reliability and predictability, and increasing productivity or throughput.

This aim of this study is to investigate the CCPM deployment at the Bosch CCTV development site regarding the CCPM rules and prescriptions, CCPM asserting benefits, important aspects for the CCPM implementation and how these are addressed, and finally provide some improvement proposals for the perceived problems. The methods used in this study were interviews, a literature review, and experiences or findings to date at Bosch CCTV development department.

The experiences at Bosch CCTV indicate that the Critical-Chain application gives more visibility or transparency in the projects status as it is asserted. CCPM addresses the existing uncertainty in the project schedule through setting aggressive schedules by removing safety time from the individual tasks and aggregating them in time buffers, which are inserted to strategic places in the project network schedule in order to capture the existing variations in the project duration.

We perceived, however, that because of incorporated high uncertainty and risks in the development projects, the allocated project buffers do not provide sufficient protection for the original promising project due dates. In the development projects, despite simple monitoring of the critical tasks with high buffer consumption rate, the CCPM prescription for swapping resources from non-critical tasks to the critical tasks for recovery means is difficult to execute. The tasks in these projects often contain non-routine activities that require the resources assigned to them to acquire necessary knowledge during the executing tasks which enables them to complete their task. This makes the flexibility of resource swapping or inter-changing of resources between the chains difficult or almost impossible.

Further in this research, the important aspects for the CCPM implementation are explored and usefulness of piloting CCPM deployment is discussed. In addition to the required changes for the CCPM implementation within the organization, we observed that ability and performance of Critical-Chain methodology has also an impact on implementing of these aspects, e.g. the organization support for CCPM deployment and maintaining of implemented changes within the project organization, because CCPM does not address the technological uncertainties and risks which highly incorporated in the development project tasks.

To improve coping with uncertainties and risks by CCPM application, we recommend including the risk management activities in the project planning and taking them in the buffer sizing calculations. In case of unrecoverable depletion of project buffer, we further recommend rescheduling the project. Rescheduling, however, provides an update of the developments in the project scope and an update of critical tasks in the course of project execution. To avoid confusion and dissatisfaction in responsibilities and activities included in each role, we suggest reaching and maintaining clear agreements between the project participants. To deal with the substitution and interchangeability

problems for human resources, we suggest not assigning all existing resource capacity to the projects. These highly trained and capable resources can then freely support the running-tasks in case of delay or interruptions and substitute the resources in case of absence. Also we suggest that other resources that are temporally idle invest their time in the identified critical tasks to increase the interchangeability of resources working on them.

The projects in each portfolio at Bosch CCTV, i.e. Camera or Digital Video Recorder are usually interdependent and share a common resource pool. To deal with these interdependencies and to decrease overloading of resources, finally we propose to separately pipeline each group of interdependent projects, i.e. with technical and resource dependencies. In case of identifying multiple constraining resources between the interdependent projects in each group, the most bottleneck resource should be chosen.

Abstract

Bosch Security Systems Eindhoven Co. (Bosch CCTV) is engaged in development projects for advanced digital video security systems, mainly cameras and recorders. Because of market competition, these developing products get increasingly more functions which lead to increasing complexity and uncertainty of the development project. To cope with this increasingly uncertainty, to improve protection of projects from delays, and to increase schedule performance and reliability, Bosch CCTV has decided to embark on the Critical Chain Project Management (CCPM) system at its development division. For product development organizations, meeting of project due dates is the vital element for successful introduction of a new product on the market. However, despite the promising benefits and many successful stories of CCPM adoption, Bosch CCTV did not achieve the main benefits of Critical Chain approach as presumed in the literature e.g., reducing project duration, gaining more reliability and predictability, and increasing productivity or throughput.

The aim of this study is to investigate the CCPM deployment at the Bosch CCTV development department regarding the CCPM rules and prescriptions, CCPM asserting benefits, important aspects for the CCPM implementation and how these are addressed and formulating proposals for the perceived problems. The first part of this thesis elaborates on the Critical-Chain rules and prescriptions and explores its differences with traditional project scheduling. Further the critical chain prescription for the multi-project environment is discussed. In the second part of this thesis, the general characteristics of the (software) development projects are discussed and the Critical Chain application for the development project at the development department is surveyed. Further the important factors for the successful CCPM implementation are explored. The outcomes are a list of the perceived deficiencies of the CCPM deployment, the encountered problems for the CCPM implementation, and providing some improvement proposals for the most significant problems.

Table of Contents

1 Introduction

1.1 Background

Product development projects often exceed their planned schedules. According to the Standish Group Report of 1999, nearly 75% of all development projects missed their target release due date or never completed at all (Standish Group, 1999). This is often related to uncertainty or contingencies inherent in development projects in which requirements, technology, skills base, business environment, and culture are in a state of flux. To cope with the uncertainty, managers and project personnel have traditionally learned to compensate by adding additional time to their schedule estimates. Yet even when they do, projects still overrun their schedules.

Achieving speed to the market is a critical success factor of today's (new) product development projects. This incorporates accelerating of the development process from concept generation to market launch. Traditional project management is not sufficient to meet the needs of the high-paced and competitive product development projects. The competition forces shorter cycle times and project planning demands better predictability in product release dates. Product development projects are inherently uncertain, resources mostly are shared between a number of concurrent projects, and tasks often do not complete on time. Project duration is considered the major constraint of projects in general (Steyn, 2002). The magnitude of the problem creates a need to find an alternative approach to how projects are planned, launched, and executed.

Critical Chain (Goldratt, 1997; Newbold, 1998) is a relatively new way of project scheduling and management. The reasons for introducing this new approach are the perceived deficiencies in more traditional project scheduling techniques. To cope with uncertainty and contingencies, estimated durations in traditional project scheduling typically include a safety time for each task. However, these safety times are often wasted because starting a task is left to the last minute (Student Syndrome) or work is expanded to fill the time available (Parkinson's Law). Critical chain reallocates these safety times in the form of buffers and strategically places them in the project network schedule to protect the whole project from delays.

Many examples of successful applications of Critical Chain methodology have been quoted in the literature (Leach, 1999; Zultner, 1998). Growing experiences with the Critical Chain methodology shows exceeding benefits across many cases (Leach, 2005):

- Increased on time delivery of project
- Reduced project duration (increased speed)
- Increased project team satisfaction, improved teamwork and focus
- Increased organization throughput with same resources (productivity)
- Increased project schedule reliability and predictability

Critical Chain Scheduling (CCS) differs from traditional project scheduling in how to prioritise task and resources and how to deal with uncertainty and the effects of Murphy's Law (if anything can go wrong, it will). It has advantages over the standard way of planning according to the Critical Path Method (CPM) but it has its own drawbacks. It is no silver bullet that solves all schedule performance and schedule reliability problems. Critical Chain methodology is simple to understand but it is also challenging to implement (Zultner, 1998). It has high potentials but these are hard to achieve due to the culture changes required throughout an organization (Devine, 2004).

Implementing CCPM is not only about installation of the software package supporting its rules and teaching project participants to use it. The necessary changes in mind-set, achieving project people support, and involvement throughout an organization are major factors and are also impediments in successful CCPM implementation. Frequent monitoring and evaluation of the change process during and after CCPM implementation help organizations to maintain CCPM rules in practice. Otherwise there is the possibility that sooner or later the adopted Critical-Chain principles and obtained CCPM buy-in (sufficient agreement to make or tolerate the changes) among project staff will be lost and the organization reverts to its old habits. The CCPM potential benefits are to be reaped if it correctly implemented and maintained. In a detailed report, Herroelen et al. (2002) state: "The critical chain scheduling and buffer management methodology has much to offer if applied wisely and if the practical implications and limitations are well understood."

The CCTV Business Unit (BU) of Bosch Security Systems (Bosch ST) is a leading producer of video surveillance equipment including Closed-Circuit Television (CCTV) cameras / monitors, switchers, control systems, Digital Video Recorders (DVRs), and the latest IP (Internet Protocol) network video products for use in security systems, which are sold in all countries of the world. Bosch ST/CCTV BU products can be found e.g. in shopping malls, casinos, and banks or airports. To create unique products both in features and performance, CCTV BU designs its own key components such as dedicated Integrated Circuits (IC's) for use in their CCTV products. All products are developed in projects that follow the Bosch ST/CCTV Product Realization Process.

In order to protect projects from delays and to increase schedule performance and reliability, the Critical Chain Project Management (CCPM) system has been implemented in the CCTV Development Division of Bosch Security Systems in Eindhoven (The Netherlands) and Lancaster (USA) at the end of 2004.

1.2 Problem Statement

Despite promising benefits and many successful stories about CCPM implementation, the Bosch CCTV Development Eindhoven has still not achieved the main benefits of Critical-Chain application e.g. gaining more reliability, productivity, and reducing the project duration. Management of CCTV takes the view that the major cause behind this is that the CCPM rules and principles are not sufficiently adhered to or not effectively utilized by the project stakeholders at the Bosch CCTV.

1.3 Research objective and questions

The intent of this thesis is to study the CCPM system at the development department of Bosch Security Systems Eindhoven (Bosch CCTV) and to explore opportunities for improvement in order to address the identified problem. To address the identified problem we formulate the following research questions:

1. What are the major differences between the Critical Chain Scheduling (CCS) and the Traditional Project Scheduling (TPS)?

2. Are there differences between the theoretical Critical Chain rules principles and the CCPM application at the Bosch CCTV development site?

3. What are the CCPM implementation aspects for deploying CCPM in an organization? How far these are achieved at the CCTV development department?

4. What areas need improvement? What possibilities are there for improvement?

1.4 Research Methodology

In order to elaborate on the rules and concepts of the critical chain project management (CCPM), and to explore the CCPM implementation factors, different reference books, papers and handouts about the CCPM theory will be reviewed.

Further, to investigate the CCPM deployment at the development division and also to gain information and explore the improvement areas, we interviewed 13 CCPM practitioners (1 senior manager, 2 Project Managers, 4 Task Managers and 6 Task Performers) at Bosch CCTV about their experiences and findings regarding the CCPM deployment at the development department.

1.5 Research Scope

This research is limited to the elaborating on the Critical-Chain principles and rules through a literature review, investigating the CCPM application in software product development projects at the Bosch CCTV development department, and exploring areas for improvement as perceived in practice. This thesis also explores the important aspects for CCPM implementation and surveys how these are addressed by the Bosch CCTV.

1.6 Thesis Structure

The text of this thesis is structures as follows: Chapter 1 introduces and formulates the problem statement. Chapter 2 addresses the first research question and reviews the critical chain scheduling and traditional scheduling methods. In chapter 3 we study the critical chain approach for the multi-project domain. Chapter 4 addressed the second research question and provides an overview of the characteristics of the software development projects and investigates the CCPM application at the Bosch CCTV development Division. Chapter 5 addresses the third research question and describes the important aspect for the CCPM implementation and perceives how these are addressed by the Bosch CCTV. In chapter 6, forth research question is addressed and some improvement proposals are provided regarding the significant encountered problems for the CCPM deployment at Bosch CCTV. Finally a summary of main conclusions and suggestions for further research are given in Chapter 7.

2 Critical Chain Scheduling versus Traditional Project Scheduling

2.1 Introduction

Critical Chain Project Scheduling (CCPS) has been introduced as an improvement to the Traditional Project Scheduling (TPM). However, key concepts of the Critical Chain methodology are readily used as an extension to the well known Critical Path Method (CPM). This chapter aims to answer the first research question "What are the major differences between the Critical Chain Scheduling (CCPS) and the Traditional Project Scheduling (TPS)?" In order to answer this question, we first give a review of traditional project scheduling methods and describe the Critical Chain Project Scheduling (CCPM) methodology. Further we describe the differences between the two scheduling techniques.

2.2 Traditional Project Scheduling (TPS)

In this section the traditional project scheduling methods and the developments for improving them are described.

2.2.1 CPM (Critical Path Method) / PERT (Program Evaluation and Review Technique)

The Project Management evolution started with the development of Gantt chart in 1917 and the development of the project scheduling techniques Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) in the late 1950's. For the first time CPM was applied in 1957 to the construction of a new chemical plant by DuPont Corporation and PERT was developed in 1958 by the U.S. Navy to help measure and control the progress of the Polaris Fleet Ballistic Missile program. CPM is a deterministic method that uses a fixed time estimate for each task while PERT is a network model that allows for randomness in task completion times. Both methods share similarities and still form the basis of the project planning and control of many projects. The key distinguishing factor of PERT as compared to CPM is the use of probabilistic task durations.

CPM/ PERT models the tasks of a project as a network, and provides a graphical view of the sequence, and relationships among the individual project tasks that are required for the completion of a project. Typically, relationships between tasks are defined as being *finish-to-start* (FS) whereby the preceding task must be finished before the succeeding task can start. The precedence technique is designed to handle other situations as well, namely start-to-start (SS), finish-to-finish (FF), and start-to-finish (SF) .These relationships are used to specify tasks that overlap to some degree. There are two graphical variations of CPM/PERT. The traditional so called (PERT) Activity on Arrow (A-o-A), or an arrow diagram, because tasks are presented in the network as arrows or lines. Another and most used approach for both PERT and CPM is Activity-on-Node (A-o-N) or Precedence diagram (Figure 2.1).

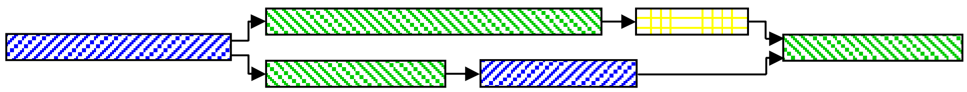


Figure 2.1 A Precedence diagram

CPM/PERT identifies the sequence of tasks (the critical path(s)) in the project and predicts how long the entire project will take. The critical path is the longest sequence of depending tasks (each dependent on the preceding one) from the begin task toward project end task in the project network that together requires the longest duration to complete the project. The critical path determines the earliest time by which the project can be completed. The significance of the critical path is that the tasks that lie on it cannot be delayed without delaying the project, which means the tasks on the critical path(s) have no slack or float.

The critical path can be identified by determining the following earliest and latest (start and finish) times for each project task in the project network schedule.

- ES- earliest start time: the earliest time at which the task can start given that its precedent tasks must be completed first.
- EF- earliest finish time: equal to the earliest start time for the task plus the time required to complete the task.
- LF- latest finish time: the latest time at which the task can be completed without delaying the project.
- LS-latest start time: equal to the latest finish time minus the time required to complete the task.

These times can be computed for all tasks within a network by making so-called Forward Calculations when calculating the earliest (start or finish) times and Backward Calculations when calculating the latest times on the project network. If in the project network, one task has more than one predecessor (more arrows coming into a subsequent task) the largest completion time among precedence tasks must be considered in forward calculations (tasks E or H in Figure 2.2). In case of backward calculation, when one task has more than one successors (more than one arrow going out of a precedent node) the smallest LS (latest start) time among the subsequent tasks must be chosen (task B in the Figure 2.3)

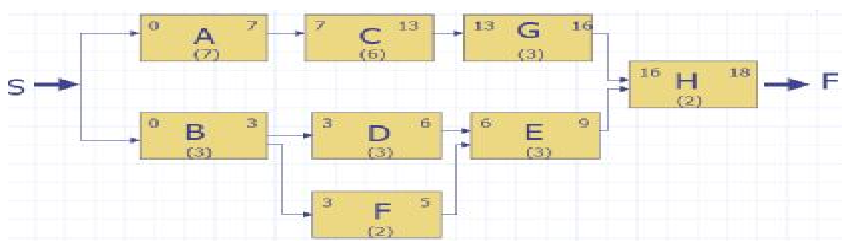


Figure 2.2 Calculating the forward pass

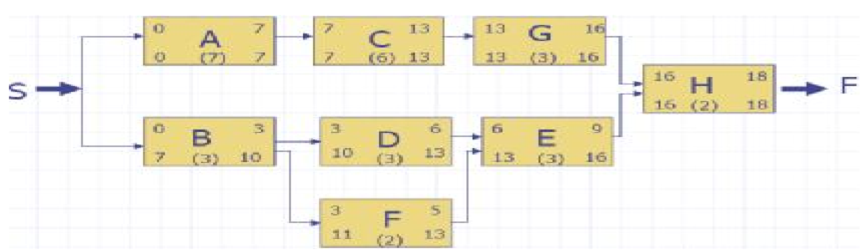


Figure2.3 Calculating the backward pass

The (total) slack or float time for a task is the time between its earliest and latest start time, or between its earliest and latest finish time. So the (total) slack is the amount of time that a task can be delayed without delaying the project. The critical path is then the path (or paths) through the project network in which none of the tasks has slack, that is the path for which ES=LS and EF=LF for all tasks on the path. Consequently, if the completion of one or more (critical) tasks on the critical path takes longer than firstly estimated durations then the whole project schedule will slip unless corrective action is taken. Because of its impact on the entire project, a critical path analysis is an important aspect of the CPM / PERT project planning method. An identified critical path in the project network diagram is shown in Figure 2.4. The tasks with bold borders are on the critical path.

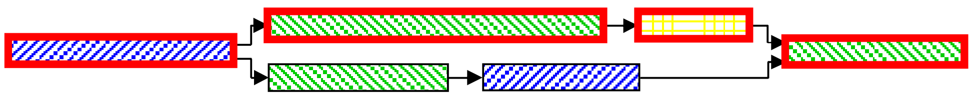


Figure 2.4 Identifying Critical Path

In CPM / PERT project scheduling, also tasks with non-zero slacks are scheduled as-soon-as-possible (ASAP) from the project start date. This scheduling places work as close as possible to the front of the schedule.

2.2.2 RCPSP (Resource Constrained Project Scheduling Problem)

The well known CPM and PERT techniques that were developed independently are based on the unrealistic assumption that the resources are unlimited (e.g. human, machine or tools, material, etc) and neglect the issue of resource contention. This assumption can lead to ineffective resource usage during the project execution and project delays. Consequently, the importance of resource availability gave rise to the subject of project scheduling under limited resources and emerging of (deterministic) Resource Constrained Project Scheduling Problem (RCPSP) thereafter (Pritsker et al., 1969).

This problem has a set of one or more limited resource types and a set of task to be scheduled. The tasks are interrelated by two kinds of constraints: precedence constraints, which indicate that certain tasks cannot be started before their predecessors have been finished, and resource availability. The execution of each task consumes both time and resources. This problem aims to determine a precedence and resource feasible baseline schedule (the original planned schedule against which the actual project execution is compared) that minimize the project duration. The well known RCPSP is a deterministic problem and it assumes that each task has a fixed duration and during its execution there is a fixed resource demand for each type of resources. The RCPSP is a generalization of job shop problem and is NP-hard which means it cannot be solved optimally in polynomial time (Blazewicz et al., 1983). The last 20 years, developing algorithms and heuristics to solve this problem dominates the project scheduling literature (for overviews: Herroelen et al. (1998); Kolisch and Padman (1999); Kolisch and Hartmann (1999); Brucker et al., (1999); Demeulemeester and Herroelen (2002)). However, the NP-hard nature of RCPSP makes it difficult to apply it in realistic sized projects. Due to this difficulty most commercial software packages (tools) often use simple priority rules based heuristics for constructing a precedence and resource dependent schedule that are hoped to be close to the optimum (Herroelen, 2006). Despite the RCPSP objective to provide a precedence and resource dependent project baseline schedule, the generated baseline schedule cannot cope with uncertainties and contingencies during the execution phase of project tasks. Because of the stochastic nature of task durations and inherent uncertainty in task estimates, project tasks may take longer or shorter than initially estimated, resources may not available when they are required and occurrence of unforeseen disruptions during task execution, etc. Hence, the validity of static deterministic scheduling has been questioned and criticized (Goldratt, 1997).

2.2.3 TPS shortcomings and need to new project scheduling approaches

Traditional project scheduling techniques were developed to plan and control large scale and complex but fairly routine projects with minimal uncertainty in the project completion times. For projects with non-routine tasks, such as product development, there is more uncertainty in completion times and scope of work. Hence, this uncertainty limits the usefulness of the deterministic CPM and RCPSP techniques because they do not consider the time variations that have a great impact on the completion time of a complex project. The uncertainty existing in every project is the underlying main cause for most schedules overruns (Rand, 2000). Recently, however, besides the need for a resource and task feasibility of a baseline schedule, stability of generated schedule has become a central point of attention in project scheduling (Herroelen and Leus, 2005a).

To cope with uncertainty, however, the traditional PERT scheduling technique, allows for randomness in task duration times. Task durations are then modeled as stochastic variables with an appropriate beta distribution, and a simple approximate method is used to calculate task durations. For each task, PERT includes three time estimates, the shortest possible (most optimistic) time the task will take, the most likely length of time, and the longest time (most pessimistic) that might be taken if the task takes longer than expected. For a beta distribution, the expected duration for each task can be approximated as the weighted average of the most optimistic, the most pessimistic, and the most likely time estimates. PERT deals with uncertainty in the same way for all tasks, whether or not they are on the critical path (Rand, 2000). Also the stochastic variant of RCPSP (that allows the stochastic task durations) while being more realistic, leads to much more complexity in the solving procedure because of NP-hard nature of RCPSP.

In traditional project scheduling to protect the completion time of each task against contingencies during the performing of tasks, safety times are included in the estimated task duration. Unfortunately during the execution phase of a project these safeties are wasted because of main focus on the due date of each task as a milestone. This approach encourages the negative behaviors of human-resources e.g. using all time that is available for executing of a task (known as Parkinson's Law) and losing of stimulant among resources for schedule savings through early finishes.

Also, the issue today is the need for a project scheduling approach that supports multiple simultaneous projects that commonly use a pool of shared resources. In a multi-project environment projects are interdependent. Traditionally projects are scheduled as if they are independent, and it is impossible to foresee how delays in one project impact other projects in an organization. In particular, companies that concurrently launch new product development projects encounter the complexities in managing multi-projects. Because of the complexity of these projects, incorporated uncertainties in the tasks and project scope, and changing priorities in multi-project environment, the RCPSP solution provides no practical and stable schedule for the uncertain multi-project environment.

2.3 Critical Chain Project Management

In order to cope with uncertainty, and its negative impacts on projects completion time and doing more projects without adding resources (multi-project management), the Critical Chain Scheduling and buffer management approach were developed. Critical Chain methodology is based on an on-going improvement methodology called the Theory of Constraints (TOC), first developed by Goldratt (1984). TOC is well established in manufacturing as a methodology for managing production planning and scheduling. It is a tool for managing repetitive production systems based on the principle that every system has a constraint, and system performance can only be improved by enhancing the performance of the system constraint (Raz et al., 2004).

In a manufacturing environment, TOC assumes that not more than one manufacturing station will be operating at its full potential capacity when the system is performing as well as possible. Using of full capacity of each manufacturing station can cause work in process inventory at the constraining station in the production sequence. The result is then increased lead time and wasted resources. Hence, the performance improvements in a system cannot be achieved through any local improvement unless the bottleneck station is relieved. Goldratt defines global system performance in terms of throughput (Throughput is the rate at which products are produced or projects are completed).

The application of TOC to project management was first described by Goldratt (1997). TOC is applied to both single project and multiple-project environments. TOC first requires that the goal of the entire system is identified. Applied to a single project, TOC identifies the promised project due date as the primary goal. In a multi-project environment, concurrent projects depend on a pool of shared resources and the objective is to maximize the throughput of projects.

The TOC approach prescribes that the system constraint has to be identified. It is then should be focused only on the identified constraint and alleviate it until it is not a constraint any more. The TOC is based on five focusing steps for global system performance improvements and applied to project management as follows:

1. *Identify the system's constraints.* TOC identifies the system's constraint as that part of the system that constraints the objective of the system. For a single-project environment this means identifying the *Critical Chain*, or "the longest chain of precedence and resource dependent tasks that determines the overall duration of a project." as the constraint. In the multiple-project environment the so called ***drum resource*** that more than any other limits the number of projects that an organization can deliver, is identified as a constraint.

2. *Decide how to exploit the system's constraints.* For a single-project this means focusing on the tasks in the critical chain to ensure that work is performed efficiently and without delays. To achieve this, the key contributors of delays in tasks completion time should be identified. Applied to multiple-project situation this means prioritizing all projects and then staggering them according to the capacity of the drum resource, ensuring that it is not overloaded.

3. *Subordinate everything else to above decision*. Applied to a single-project this means that non-critical tasks must not interfere with or delay work on critical tasks. In multiple-project environment this means that non-critical resources may wait to ensure high utilization of the bottleneck resources across the projects.

4. *Elevate the system's constraints.* For both single and multiple-project cases this step suggests investment in additional resources, or increasing the capacity of resources that most impact the critical chain or total project organization throughput. In certain cases, elevating of critical chain constraint may be carried out by assigning some of the non-critical resources to the critical chain tasks.

5. *If, as a result of the previous steps, the constraint has alleviated, return to Step 1.*

In traditional CPM/PERT project scheduling, resource availability is not taken into account and resource allocation is done as an additional step. Resource dependencies, compounded with task dependencies, further decrease the probability that a task will finish on time. In order to obtain a realistic and stable baseline schedule, the RCPSP, however, takes resource availability into account to the extent that tasks done by same resource type with limited capacity, if it's not possible to perform in parallel, are scheduled and performed in series. CCPM defines the critical chain (see Figure 2.5) as the set of tasks which determines the overall duration of the project, by first solving the RCPSP and taking into account both precedence and resource dependencies (availability of resources) when creating a project base line schedule. The CCPM identifies the longest chain of both precedence and resource dependent tasks in the generated project schedule as the critical chain of project network schedule. If resource availability is not a constraint, then the identified project's critical chain is identical to critical path of CPM method.

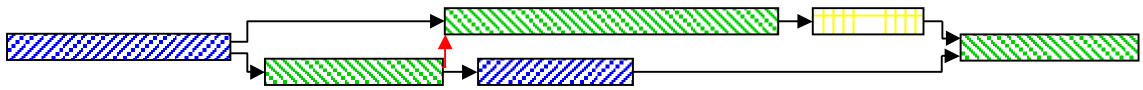


Figure 2.5 Solving of the resource constraint project scheduling problem (RCPSP) before identifying of critical chain in CCPM

The traditional Resource-Constrained Project Scheduling Problem (RCPSP) is essentially the same in the Critical Chain Scheduling. However explicit attention to the presence of uncertainty, protecting of generated (deterministic) precedence and resource dependent base line schedule (during the project execution) through inserting of buffers, are the key difference between the two approaches.

The CCPM approach does not prescribe a specific RCPSP algorithm out of the numerous algorithms that have been published in the literature that vary in terms of average distance from the optimum. Consequently, the identification of the critical chain(s) is entirely dependent on the procedure used for generating the baseline schedule (or solving of the RCPSP). Depending on the used RCPSP algorithms or heuristics for constructing Critical-Chain (resource constrained) schedule, the sequence of the resource-dependent tasks may differ in the resulted Critical-Chain schedule. The resulting Critical-Chain schedules from the most known CCPM packages ProChain, Concerto and PS8 show differences in the sequence of some resource-dependent tasks and the durations of the resulting-CCPM schedules because each CCPM tool uses different simple priority rules for dealing with RCPS problem (Cerveny, 2005). Goldratt referred to this issue, claims that in each case “the impact of scheduling method used is seldom larger than the uncertainty of the project” without defending it (Coldratt, 1997, p217).

In the literature, the Critical-Chain schedule is also referred to as resource leveled CPM schedule. The resource leveling (leveling of resource usage over the already obtained project due date by CPM) is different than solving the resource-constrained project scheduling problem (minimizing the project duration subject to the precedence and resource constraints). Hence it should be noted that the generated CCPM schedule is not the same as the resource leveled CPM schedule (Herroelen and Leus, 2005b, p105).

Resources demanded by the tasks on the critical chain are critical resources. However, the CCPM method resolves all anticipated resource constraints while determining the project critical chain. Because at least some of the resources have limited availability, the resulting baseline schedule is likely to be longer than the schedule obtained with basic Critical Path Method, as critical tasks are sequenced while waiting for the resources they require.

Consider two tasks that use the same limited resource, such as the back-cross-hatched tasks in Figure 2.5. They are scheduled in series because they share the same resource. This requirement expands the project duration. The critical chain is identified as the chain of precedence and resource dependent tasks that determine the overall duration of the project.

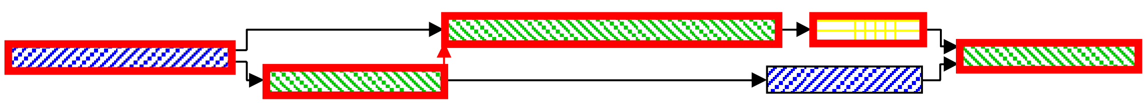


Figure 2.6 Identifying of Critical Chain and setting of non- critical tasks ALAP

In the constructed critical chain schedule the non-critical tasks are scheduled as-late-as-possible (ALAP) based upon the target due date (Figure 2.6). This approach provides advantages similar to those offered by ‘just-in-time’ (JIT) in a production environment. These benefits include minimizing work-in-progress (WIP) and not incurring costs earlier than necessary. Also, by scheduling tasks as-late-as possible, more knowledge is available and will minimize the need for re-work when possible changes in the scope of the task imply a higher risk of rework of tasks that are started ASAP. Less rework will also result from the fact that the task performers simply have better information about their task mission. Finally, starting work ASAP creates countless opportunities for multi-tasking. The main drawback directly related to the ALAP-scheduling is that, in traditional critical path terminology, all tasks become critical. An increase in duration of any task will push out the project end date. As will be explained in Section 2.3.5, buffers are inserted at key points in the project plan to act as shock absorbers in order to protect the project end date against variations in task duration. In this way, we can exploit the benefits of ALAP-scheduling with adequate protection against uncertainty (Steyn, 2002).

2.3.1 Variation and Uncertainty

In the context of project management, variation concerns the inherent uncertainty of task durations. The duration of any task will vary, depending on various reasons. Those reasons could be due to common cause variation (normal random fluctuations) or special cause variation, which could be identified, i.e., a cause that is specific to some group of workers, to a particular production worker, to a specific machine, or to a specific local condition. However, even when each and every step of a task is executed within acceptable duration, the project as a whole will still contain a degree of variation. This is referred to as common cause variation. The common causes are variations in duration that predictably occur because they are part of the system within which projects are performed. Special cause variation refers to the variation in parts of the task process, which essentially makes those parts of the process unpredictable. Special causes are dealt with as part of risk management (Leach 2005, p104).

Traditional project scheduling accounts for the presence of inherent common cause variation through adding safety time (the amount of time needed to ensure on time task completion) to each project task, whether they are on the critical path or not. A project manager’s job then consists of making decisions and taking actions based on the finish date of each task and how it affects the overall project schedule. This practice makes project management and control difficult. CCPM instead offers an alternative, in order to manage the variation, resulting in shorter schedules. It improves the accuracy of prediction (reliability) for project plans by addressing variation through the use of time elements called buffers, placed in strategic positions in the project schedule. These buffers aggregate the protection (by removing of safety from the individual tasks) that a project needs to meet its due date and allow focus on project duration (Leach, 1999).

2.3.2 Negative Resource Behavior

Traditional project scheduling models rely on tasks with embedded safety times and task due dates to schedule and control projects. This approach does not take into account the impact of negative resources behaviors that will minimize the ability to gain time on the schedule. However, as discussed above, project tasks are likely to take much longer than expected, and less likely to take a shorter time.

The Critical Chain methodology is not only a scheduling tool but also provides managing guidelines in order to deal with the human (resource) nature that can adversely affect the duration of task execution. To this end, the CCPM methodology uses an aggressive project schedule with shortened task duration estimates based on the idea of correcting such behaviors and focuses on human problematic behaviors during performing tasks in a project. Traditional project scheduling methods stimulates these negative behaviors because task performers are aware of embedded safeties of task estimates and tend to commit only on the estimated duration and due date of each task. Examples of these behaviors are:

Parkinson's Law: "Work expands so as to fill the time available for its completion" (Newbold, 1988). This can happen when a resource is unsure of the completion criteria and continues to work a task often to a perceived due-date. This can also happen when a resource hits an obstacle to task completion. As the obstacle delays the completion of the task, the resource may work on other aspects of the task, thereby expanding the task-content to fill the time it takes to get the obstacle resolved. The best course would be to focus on the obstacle, get it resolved, and then finish the task. The negative behaviors are explained below are also other examples contributing to Parkinson's Law.

- Gold-plating: This is including enhancements that are not necessary to accomplish the objective of a project's task. This is a significant factor in software development projects. The critical chain helps to minimize "gold plating" when a task performer who is on the critical chain is aware that taking time to add features will delay the project completion. This consciousness encourages the task performers to complete their task as soon as possible.

- Three-minute egg rule: This happens when there is a belief that the quality conditions are not to be met if the task is completed in shorter time than the estimated duration time for it or the assumption that quality conditions requires using the entire time allotted to execute a task.

- Sandbagging: This is an American slang term that implies inflating an estimate. If resources finish their or her task earlier than planned, they might be accused of sandbagging the estimates instead of being rewarded for completing ahead of schedule. In such an environment, resources are worried about the future estimates that will be cut by management. Hence, they enjoy keeping the early finishes hidden and officially finish on schedule.

Multi-tasking: Another reason that cause tasks take longer is multi-tasking. Multi-tasking occurs when the project organization assigns a resource to perform multiple tasks during a particular time window. As a result they need more time to complete their tasks or may start their tasks later than planned. Multi-tasking can be done on the same or on different projects. This does not happen often in construction and engineering projects, because it is physically impossible. In IT projects the temptation for multi-tasking is strong because they are human-resource intensive projects. Sometimes multi-tasking can contribute to the proper execution of task. Hence the objective is to avoid *bad multi-tasking* that causes delaying of tasks.

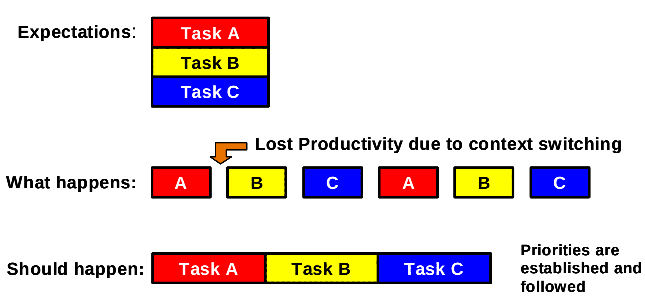


Figure 2.7 An example of multi tasking and lost productivity because of it

Student syndrome: Resources sometimes start tasks later due to the Student Syndrome or procrastination, like a student who waits until the last minute to complete an assignment for which more than adequate was originally considered. This general tendency also exists in many of human-resources involved in a project. Often, the major difficulties in completing a task are not discovered until the resource really performs his or her assigned task. This usually happens in the latter part of task execution and it is then impossible to complete the task within the time estimate.

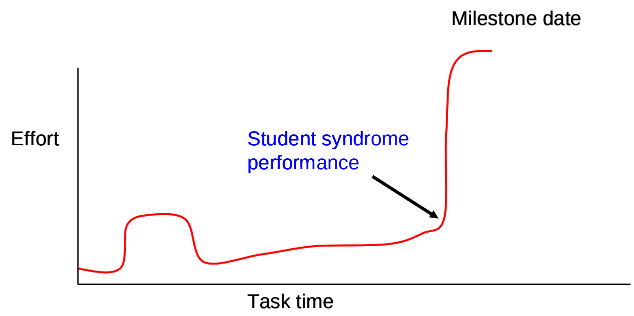


Figure 2.8 Student syndrome or procrastination
(Source: Avraham Goldratt Institute)

De-Synchronization or Dependencies effect: One reason that tasks are not completed earlier than planned is the effect of dependencies. This can happen when resources are not available when expected to work on a task and/or when task completion criteria have not been fully met from predecessor tasks. Both situations drag out task accomplishment (Kendall, in Kerzner 2003, ch.22)

CCPM confirms these human problematic behaviors hidden in each task and assumes that resource behaviors can be modified. To minimize the mentioned resource behavioral problems and to avoid wasting of allotted safety times in the duration estimates of project tasks, CCPM recommends that task duration estimates should be shortened and safeties embedded in duration estimate of each task should be eliminated. CCPM then aggregates the amounts of safety time in the form of buffers to protect due-date performance, and to avoid wasting this safety time because of bad multitasking, student syndrome, Parkinson's Law, and poorly synchronized integration. Besides, conditions should be provided wherein the baseline schedule with shortened task duration estimates, can be protected. In the next section, these conditions will be further described.

2.3.3 Aggressive task durations estimates

CCPM is based on the premise that uncertainty in task duration is the major factor affecting the ability to complete the project on time and strives to better manage the safety times that are added to each task for dealing with uncertainties.

Traditionally, planners protect tasks completion by adding safety times to task duration estimates. These safeties must deal with the uncertainty involved in the work (uncertainty embodied in Murphy's Law: if anything can go wrong, it will). Unfortunately in practice most of these safeties in different manners as mentioned above are wasted. Goldratt also refers to this matter and argues that the main reason for project overrun is because of the misuse of the safety time created within the estimated times for each task. He believes also that a consequence of the three time estimates used in PERT and their weighted mean being used for scheduling by CPM, will be a tendency to overestimate the times to give a reasonable degree of certainty of timely completion (Rand, 2000).

In traditional project scheduling, tasks duration estimates are such that the probability of timely completion is about 90%. To avoid negative behavioral problems CCPM removes the embedded safety time from the task estimates and shortens them to the point where the probability of timely completion for task is about 50%. The difference between an estimate with 90% confidence and an estimate with 50% confidence is then the removed safety time of each task (Leach, 2005). The Critical Chain methodology requires that the schedule be built with the 50% estimates without any safety. The reduced task duration is the time that a task expects to take, if a full sustainable level of effort to be made to perform it, the required resources are available when needed, and there are no significant problems during task execution (Patrick, 1999).

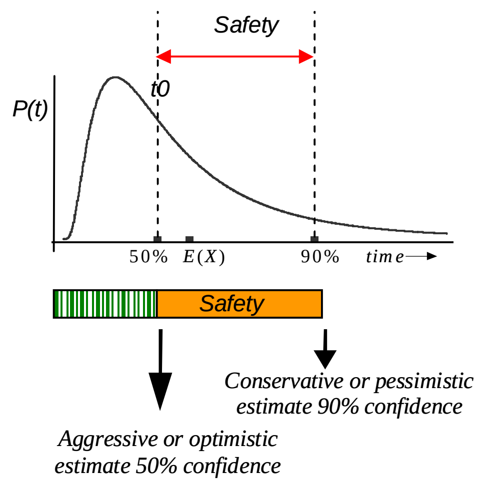


Figure 2.9 A typical probability distribution for a task with uncertain distribution.

Figure 2.9 illustrates a typical probability distribution for a task with an uncertain distribution. The horizontal axis shows time, the vertical axis shows the probability of task completion at a given time. Time *t0* is the median or the time it takes to complete the task with 50% certainty. The long tail at the right side of curve reflects the time required for the task to complete with a high degree of certainty. Goldratt (1997) suggests using the median, while the Product Development Institute (1999) argues in favour of the mean. Herroelen and Leus (2001) concluded from their full factorial experiment that the use of the mean task duration provides the safest estimates of project duration. Figure 2.10 illustrates an aggressive CCPM schedule with shortened task duration estimates (with 50% certainty of completion).

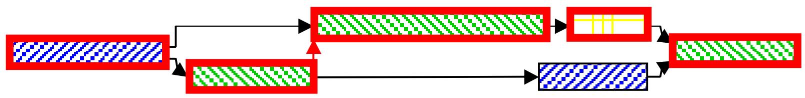


Figure 2.10 Shortening of tasks duration from 90% completion certainty to 50% certainty

2.3.4 Relay-Race (Road Runner) Behavior

To make 50% task duration estimates achievable, a further principle of CCPM is that the start and finish dates of individual tasks are not monitored during project execution. This is done to remove the pressure on task performers that work on the critical (chain) tasks and to promote acceptance of the idea that one half of the time tasks can overrun their estimated durations. In these situations, task performers do not need to wait for the scheduled start date of each task on the critical chain and each successor (critical) task can be started as soon as its predecessors are completed. This is known as Relay Race work ethic or behavior - start immediately after receiving the baton. Once started, finish as soon as possible and pass the baton to the next process (Lechler et al., 2005).

In this way it is possible to take advantage of early finishes of tasks and to compensate delays of some of late tasks. Task performers that work according to the relay race are assured that in case of reporting early finishes, their next task durations will not be shortened. CCPM strives to substitute the negative resource behaviors with the relay racer behavior by encouraging resources to focus on one project task at a time, and passing on their completed work as soon as possible. CCPM counters the Parkinson's Law by using 50% task duration estimates, not revealing scheduled task start and finish dates, and using frequent schedule updates. It counters the student syndrome through aggressive task duration estimates, providing resources prioritized task lists, and frequent status reporting.

Also because of setting of 50% probability of successful completion for each task estimate, 50% of the tasks are expected to be delayed. The primary emphasis of CCPM is for resources working on tasks to work as efficient as possible to achieve the aggressive scheduled task duration. For being late, no penalties are given if best efforts by task performers are made to avoid it. The expected delays can be absorbed by aggregated time buffers made from the removed tasks' safety that are placed in strategic positions of the project schedule. The operating mechanism of time buffers in critical chain schedule will be described in the next section.

2.3.5 Buffer Insertion

The critical chain approach suggests the shifting of focus from assuring the achievement of individual task estimates (sub-optimisation) to assuring the only project completion due date that is the global goal of the project. To protect the project due date, and to avoid wasting task safety times, CCPM prescribes that safety times should be eliminated from the individual task duration estimates and aggregated in the form of time buffers at strategic locations in the baseline schedule because CCPM claims that an aggregation of task safeties in the form of buffer provides a better protection regarding single task safety times (Herroelen et al., 2002). Leach (2005) states "The Critical Chain methodology exploits the statistical law of aggregation by protecting the project from common-cause uncertainty of the individual tasks in a task path with time buffers at the end of path in the project network" (Figure 2.11). The protecting time buffers are not slack time. They are an integral part of critical chain schedule with shortened or aggressive task duration estimates. In addition, estimates of durations are never perfect. Without inserting of absorbing time buffers the protecting of project completion due date and committing to it in the critical chain planning cannot be reliable.

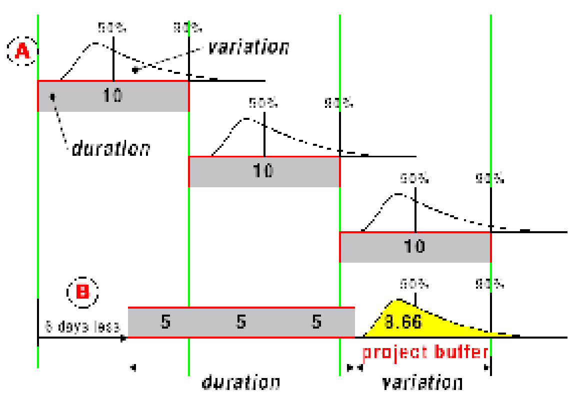


Figure 2.11 Achieving 90% certainty with an aggregated Project Buffer (Source: Zultner, 1998)

CCPM shifts the safety times associated with the critical chain tasks to the end of the critical chain in the form of a *Project Buffer* to protect the project due date promised to the customer from variation in the tasks of critical chain (figure 2.12). This improves the reliability of the overall due date as well. The promised delivery date of the project is then the sum of the critical chain duration plus the Project Buffer duration.

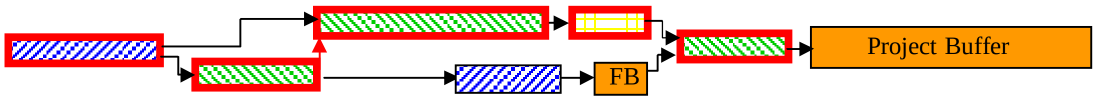


Figure 2.12 Inserting of Project and Feeding Buffer(s) to protect shortened project schedule

Feeding Buffers are another type of time buffers in CCPM that are inserted whenever a non-critical chain task joins the critical chain (figure 2.12). Their aim is to protect the critical chain from unforeseen difficulties and disruptions on the non-critical tasks feeding it, and to allow critical chain tasks to start early in case things go well. When the project network consists of only one path, no feeding buffers are needed.

There is a possibility that the insertion of a feeding buffer into a non-critical chain make the resulting feeding chain longer than the critical chain. This means that non-critical tasks have to be started before the first task on the critical chain or that time gaps need to be introduced into the critical chain of buffered baseline schedule. To answer this problem, the critical chain is simply defined as the longest chain of resource and precedence the un-buffered CCPM project schedule before the buffer insertion (Herroelen and Leus, 2001).

The third type of buffer used by CCPM is called a Resource Buffer, which is a virtual task inserted prior to critical chain tasks that require critical resources. Critical resources are resources that work on the critical-chain tasks and should not be interrupted or multi-tasked during their performance. The resource buffer works as an advance warning signal to the critical resources that should work on a critical chain task which is expected to start shortly. According to CCPM, this wake-up call will cause the critical resource to complete any non-critical task and be ready to start work on the critical chain task as soon as its predecessors are completed. The resource buffer does not use time on a schedule which it protects the critical chain from lack of availability of required resources and provides the possibility for critical chain tasks to start early (Leach, 2005).

CCPM recommends that the buffered baseline schedule and the identified critical chain should not change during project execution except for major disruptions that consume in high speed the protection offered by the project buffer. This is encouraged by the idea that rescheduling and changing the critical chain may lead to losing focus.

2.3.6 Buffer Management

In CCPM, in order to provide focus and be proactive during project progress, the buffers are monitored to ensure that critical chain and project due date are protected. This mechanism is called Buffer Management and is the key to managing and tracking of project performance in CCPM. Every task in a critical chain scheduling is connected either to a project buffer or feeding buffer. As project execution proceeds, if a task takes longer than estimated, it consumes the buffer that its path is connected to.

The buffers help management to act proactively. Buffer Management identifies potential problems much earlier than they would ordinarily be detected using traditional project management techniques. Buffer Management frequently compares the consumption rate of each buffer to the progress rate of the (task) chain leading to it. As long as there is some predetermined proportion of buffer remaining, the chain progress (project progress in case of project buffer) is assumed to go well. If the executing chain consumes a buffer by a certain amount, a warning signal is raised. If the buffer consumption rate is high so that whole of the buffer are expected to be consumed before completing of the tasks leading into it, corrective action should be taken.

The buffers are divided into three equally sized regions (Green, Yellow, and Red). If the buffer consumption is in the green zone, no action is required. If the consumption enters the yellow zone, then the project manager should assess the problem and think about possible courses of action. If the buffer deletion reaches the red zone, then the project manager should act. Figure 2.13 illustrates the three zones associated with a buffer.

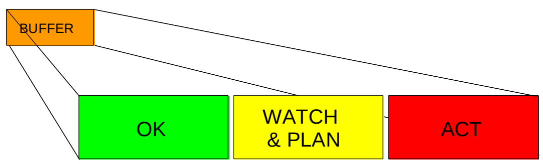


Figure 2.13 Buffer Zones Green, Yellow, and Red (Source: Goldratt, 1997)

The two essential measurements of project performance in buffer management are the percentage of the critical chain completed and the amount of the project buffer consumed. The relationship between the critical-chain completed to the amount of project buffer consumed is the signal to management for appropriate action. Project review meetings focus on whether the completion of the critical chain is at a pace for completion without consuming the project buffer. In this environment, the role of the project leader shifts from a focus on all tasks to those tasks that are on the critical chain. Also, focus remains as necessary for any feeding chains that may be in danger of impacting the ability to start a task on the critical chain.

The tri-colored graph located below is called a Fever Chart. This chart shows the % Buffer Consumed vs. % chain complete for a (single or multi) project. The purpose of the Fever Chart is to instantly tell the project status to the project manager, the client, the project team, and senior management. Measuring the trend of buffer consumption improves the early-warning aspect of buffer management.

If the trend line enters the red zone (Last third of the buffer- Figure 2.13), the project is in trouble. The Project Manager must then prepare to intervene. The project stakeholders must meet to discuss immediate corrective action that is required for the project. If the trend line is in the yellow zone (middle third of the buffer) the project manager must assess the project status carefully. Recovery action may be needed or considered. If the project is in the green zone (first third of the buffer), all is well with the project. If the project finishes in the green zone, most likely the schedule had too many safeties injected and was not properly scheduled using CCPM techniques.

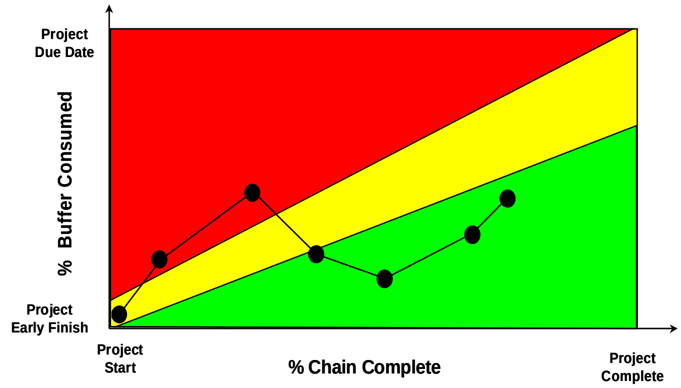


Figure 2.14 Measuring the trend of buffer consumption using Fever Chart

Figure 2.14 is a so-called *Fever Chart* for a single project. The Fever Chart provides an instantaneous snapshot of the project status. By measuring the percentage of buffer consumption relative to the percentage of the critical chain completed, the management is able to measure the status of the project at any given time. The trend line in the figure 2.11 indicates that at the beginning of project execution the project buffer is being consumed rapidly relative to the amount of the critical chain tasks accomplished.

To calculate the buffer consumption rate and project progress, frequently reporting the Remaining Duration of each task is needed. Compared to the traditional project monitoring, this is a shift from focusing on percent of work (task) complete to focusing on how much time is left to accomplish unfinished (chain) tasks. This traditional manner of project control is subject to different interpretations. There is a human tendency to say that a task is 90% completed very quickly, and then spends just as long or longer finishing up the last 10%. Hence CCPM tracks progress through the team members estimates of remaining duration rather than work performed. A comparison of the remaining duration of a task with its original estimated duration is an effective measure of progress of the tasks on the chain. The buffer management reports the remaining duration of the project buffer as measurement of the project performance.

In the Fever Chart two factors are frequently calculated: Chain Percent Complete and Buffer Percent Consumption.

- Calculating Chain Percent Complete for Critical Chain:

% Chain Complete = (original Critical Chain duration – remaining duration of longest chain) / original Critical Chain duration

- Calculating Chain Percent Complete for Feeding Chains:

% Chain Complete = (original duration of longest feeding chain – remaining duration of longest chain) / original duration of longest feeding chain

Example:

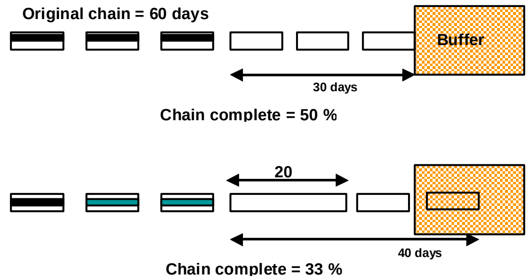


Figure 2.15 Calculating of Percent Chain Complete in Buffer Management
(Source: Realization Technologies Inc.)

- Calculating Percentage of Buffer Consumption:

The percentage of buffer consumed (% Buffer Consumption) for a buffer is the consumption rate by the chain leading to it. If there are multiple chains, the worst value is taken.

Example:

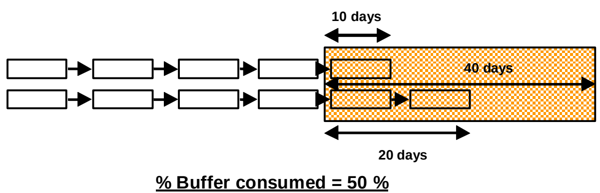


Figure 2.16 Calculating of Percent Buffer Consumed in Buffer Management
(Source: Realization Technologies Inc.)

2.3.7 Task Management

The necessary changes in the behavior of task performers (see Section 2.3.4) requires finding new people or defining new functions, e.g. task management in order to support the optimal utilization of the Critical-Chain rules. Active and close managing of tasks during project execution ensures that tasks are performed with the right priority, with minimal interruptions during execution and in an efficient manner. In order to achieve these objectives, task managers use the outputs of the buffer management analysis and monitor the remaining duration of executing tasks. They coach task performers in their task mission and play a key role in minimizing multitasking, student syndrome, gold-platting, and maintaining Critical-Chain rules during the project execution in order to keep the project on track (Leach, 2005). The process needs also to protect task performers from micro-management (a management style where a manager closely observes or controls the work of their employees) while allowing the work to be performed with a relay race mentality. This keeps resources productive instead of keeping them constantly busy. Otherwise, without monitoring and managing of tasks, allocated buffers get wasted which creates the feeling that aggressive task estimates are unrealistic. This causes that the organization loses gradually its CCPM support and reverts to its old manner of working, i.e. adding safeties to project individual tasks, missing insight in the setting of priorities and dropping back into the old behavior patterns. The identified critical chain tasks and their buffer status are the focusing point for managing of tasks until they are not critical anymore or are completed.

In CCPM, one of the key changes is how people are measured and managed (Leach, 2005). CCPM points out that uncertainty in the projects exists, which implies that there is no direct way of measuring if a person is under-performing or is faced with uncertainties. Also through imposing aggressive task duration times, it is expectable that not all of tasks are performed within the planned estimates of the durations that were made. At the same time, there is a process in place that builds accountability for how resources are used. Therefore, in CCPM the resources, e.g. task performers are asked not to report their work progress in percentage of completion since this assumes fixed task durations and focuses on the amount of work done. In the CCPM monitoring and control process, the constant question is the amount of work that is left to do for each task (project), in other words, how many days are still needed to finish the task (project). The reason behind this question is the fluctuations during the task execution depending on the severity of obstacles that are encountered. Hence, in the Critical-Chain approach, the executing-tasks of project are frequently updated whenever tasks are started, finished or are still in progress (not completed yet). Task Manager uses updated buffer reports to decide which tasks to focus on next, and to determine when and where to take action to recover if the tasks are in the red zone. Issue resolution is other important task of the Task Managers. Issues are areas of concern that impact the reaching of a task (project) completion.

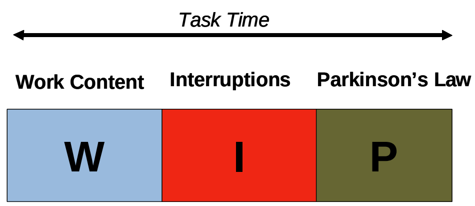


Figure 2.17 Needed total time because of execution losses

Figure 2.17 illustrates the time needed while working on a task. There is time for doing the work content, and time allowed for interruptions (waiting for issues to be resolved, decisions to be made, priorities to be clarified, waiting for other tasks to complete, multi-tasking on other higher priority work, etc). Finally, there is time consumed by Parkinson's Law (work expands to fill the time available). CCPM recognizes that actual task performance times include common-cause variation, e.g. as outcome of Parkinson's Law, and task performers are not criticized for their task-duration performance.

2.3.8 Buffer Sizing

In the literature, different ways for sizing of buffers in Critical-Chain Scheduling are mentioned. One of these was given by Goldratt (1997) who suggests that the project buffer size should approximately be half the size of the sum of the preceding tasks (with aggressive or median time estimates). Similarly, the sizes of feeding buffer should be 50% duration of the feeding chain leading into it. This method is called *Cut and Paste Method* (C&PM). The reasoning behind this method originates in statistical rules to combine variances, which means that it is possible to protect a chain of tasks to the same level of probability with much less total safety time than the amount of protection needed for each individual task. In other words the overall variance of the Critical Chain will be much less than the addition of all the individual variances for each task. Consequently, aggregation of the safety times dramatically reduces the overall estimated time for chain of tasks.

However this method is simple, it is a linear process, which means that as the size of the buffer increases linearly with the length of the chain with which it is associated. Consequently, the result may be an unnecessarily large amount of protection, which could lead to uncompetitive promised due dates and loss of business opportunities. In addition, due to the linear model upon which the cut and paste method is based, short chains of tasks tend to get dangerously short buffers. Herroelen and Leus (2001) reported in their study that the Cut and Paste method seriously overestimates the buffer sizes.

In general, using the Cut and Pate Method for project environments such as new product development are not recommended (Tukel et al., 2006; Product Development Institute, 1999). Leach (2005) recommends in each case, the size of the buffers should never be less than 25% of the time (length) of the (critical) chain leading to it, and the project buffer are typically sized between 30% and 50% of the critical chain duration. He argues that chains with many tasks of uniform length may calculate a relatively small buffer, providing insufficient protection.

The alternate buffer sizing method is known as the ***Root Square Error Method*** (RSEM). This method requires both safe and aggressive task duration estimates for each task. The first estimate should include enough safety to protect the execution of tasks against the contingencies. The second estimate is the average estimate without safety time whereby the task will be performed at a full level of effort without interruptions imposed by external factors. The RSE method uses then the difference between safe (90% probable time) and aggressive (50% probable time) estimates to calculate the most likely error in the duration of entire chain. It sizes a buffer as the square root of the sum of the squares of task safeties for each chain of tasks. In this method it is assumed that project tasks are independent. In Figure 2.8, as an example, the critical chain plan uses the 50% probable time to create a critical chain of the three tasks that is 15 days long. The project buffer, the square root or the sum of the squares of the differences between the 50% time and the 90% time (each 5 days) is 8.66. The total project duration is then about 21 days and that is 6 days less than the project duration in the traditional project planning. Tukel et al. (2006) pointed out that compared to C&PM; RSEM has a distinct advantage of not generating very large or very small buffer sizes based on the length of the feeding chain.

The methods explained above do not incorporate the project nature, e.g. resource tightness (the ratio of total resource usage to total resource availability) and complexity of project network schedule, into account when determining buffer sizes. Tukel et al. (2006) state “When in a project the total resource usage is close to the total resource availability, it is more likely that delays will occur. Thus there should be larger buffers to absorb the delays. Similarly, for a given number of tasks, as the number of precedence relationships increases, it is again more likely that delays will occur. In this case the tasks are more interrelated and a delay in a task completion will influence all of its successors. Therefore, as the number of precedence relationships increases, the buffer size should also increase.” To address these issues they suggest new adaptive methods for the sizing of buffers.

2.4 CCPM Critiques

Although the Critical Chain methodology acts as an important eye-opener to traditional way of project scheduling and control, with the idea of setting an aggressive precedence and resource dependent schedule and protecting it against uncertainty by means of inserting buffers. However, there is still a lack of consensus in the CCPM over the underlying assumptions and the availability of supporting empirical evidence.

CCPM suggests reducing of the task duration estimates by a certain percentage, e.g. 50%. Raz et al. (2004) refer to this issue and state “Clearly such an approach is problematic, not only due to the need to justify the percentage reduction chosen, but also due to the fact not all people overestimate by the same amount. At any rate, the behavioral aspects of identifying the precise amount of safety margin and taking it away from the task owner are dealt only superficially by CCPM literature, and still require empirical support”.

CCPM, however, does not provide a scientific basis for determining the buffer size, while the sizing of buffers obviously has a significant impact on the CCPM project due date (Raz et al., 2004). Therefore, it is confusing when no sound guidelines or prescriptions for the amount of cutting in original estimated task duration that has a direct impact on the project performance, are given. However, despite the different suggested methods in the literature, (Section 2.3.8) there is still a lack of consensus for the amount to be cut in original estimated task durations. Furthermore, determining the size of a buffer in a highly uncertain and competitive environment, such as the product development, is more crucial than in construction projects where the majority of tasks are routine and with less uncertainty.

The CCPM theory does not prescribe a specific algorithm or heuristic for solving RCPSP, out of the numerous methods that have been published in the literature and that vary in terms of average distance from the optimum. Creating a precedence and resource dependent base line schedule that minimizes the project duration is not easy because the RCPSP is NP-hard. Different sub-optimal procedures for solving a RCPSP may yield different schedules with different critical (chain) sequences. CCPM minimize this issue by saying that the impact of the scheduling method used is seldom larger than the uncertainty of the project (Herroelen and Leus, 2001). Also if there is more than one critical chain in the schedule, CCPM recommends choosing one and buffering the others, but does not say the reason behind this.

Another critique is that the CCPM schedule may not be stable during the project execution. However, uncertain events during the project execution-task delays, the necessity to insert new tasks, unavailability of resources may sometimes dramatically change the composition of the critical sequences. In this situation the project buffer cannot absorb completely the schedule changes and it is not sufficiently protective. Despite the CCPM dissuasion to reschedule frequently, CCPM seems not adapted to environments with very high uncertainty such as new product development, where uncertain evolution structure of the projects certainly requires frequent rescheduling (Herroelen and Leus, 2004). Also, with inserting feeding buffers, non-critical chains push back in time. This may lead to new resource conflicts and changing of the already solved RCPSP in un-buffered projected schedule. The literature is not clear about how this can be solved (Herroelen and Leus, 2004).

Herroelen and Leus (2001) have validated the working principles of CCPM. They reached the conclusion that the Critical Chain methodology acts as an important eye-opener but constitutes a serious oversimplification of the real problem and induces the need for additional research. Even if CCPM is simplistic and oversold, it is worth studying for its several pieces of good advice, in this respect, CCPM:

- Accounts for duration uncertainty by making buffers explicit;
- Explicitly addresses the resource availability;
- Focuses on the critical tasks and resources;
- Provides a proactive tool for project monitoring by managing inserted buffers;
- Provides advance notice of upcoming work to critical resources;
- Takes advantage of early finishes;
- Provides visibility to the schedule when a project is in trouble;
- Addresses explicitly the problematical aspects of human-resource behavior.

2.5 Summary

The major difference between traditional and critical chain scheduling is in how uncertainty is managed. In traditional project scheduling, uncertainty is managed by adding safeties into task durations, starting tasks as early as possible, multi-tasking and focusing on meeting task due dates. While in Critical-chain scheduling, uncertainty is managed by setting aggressive task duration estimates, generating a precedence and resource dependent baseline schedule by solving RCPSP, scheduling back from the date a project desired to complete (scheduling tasks as late as possible), inserting aggregate buffers at key points in the project network schedule in order to protect the project due date and the critical tasks against duration increases and variation, and using buffer management to monitor and control the project execution.

Critical Chain suggests the shifting of focus from assuring the achievement of task estimates and intermediate milestones in the traditional project scheduling to assuring the only date that matters and that is the final promised due date of a project. The critical chain scheduling method requires the elimination of task due dates and milestones in the CCPM schedules, and differs from the traditional project scheduling in that it is also based on dynamic schedules that are robust against uncertainty and disturbances.

In the traditional CPM method, the critical path is the longest path through the network diagram and has no slack or float. This longest chain of tasks based upon task dependencies while in the CCPM planning the critical chain is the longest chain of tasks that consider both task dependencies and resource dependencies. It is not necessarily equivalent to the longest chain in project network since, sometimes, there are non-critical tasks that begin before the critical chain tasks. This happens when the feeding buffers are inserted by pushing back their feeding chain. The critical chain, the important chain of tasks that is supposed to determine the duration of project, is started later than non-critical chain task. Also gaps may be created in the critical chain of buffered critical-chain schedule. The critical chain is thus the longest chain before the buffers are inserted in the project schedule.

There can be more than one critical path if the lengths of two or more paths are the same. The critical paths may change as the project progresses. This occurs when other paths experience delay, and redefines the longest (zero float or slack) path to complete the project. The critical chain however does not change during project performance. If there exist more than one critical chain, an arbitrary choice is made. The critical path often changes during execution because there is no buffer to absorb the variation in task durations. The Critical Chain schedule and the Critical Chain itself do not change throughout the life of the project, because the buffers absorb the uncertainties in task duration.

In the traditional project management, the negative behavioral aspects of resources are not explicitly addressed while Critical Chain methodology addresses them and strives to substitute a dedicated behavior (relay race mentality) wherein people are not measured and are not held accountable for task delays. Managing tasks by due dates is not done, and resources are asked to pass on their outputs to the next resource as quickly as possible.

3 CCPM for the multi-project environment

3.1 Introduction

Nowadays in project environments many projects occur concurrently and compete for the common critical resources. As a result managers have to constantly monitor and react to contingencies during execution of interdependent projects within the organization, but at the same time they do not have sufficient visibility of how problems and decisions on one project impact other projects. This chapter reviews the Critical-chain approach in multi-project environments and how it addresses the interdependencies of projects.

3.2 CCPM in multi-project environment

The CCPM approach provides a prescription in both, single-project domains by identifying critical chain of tasks, and multi-project environments by staggering projects based on the identified most critical resource between multiple projects.

In single-project environments, projects are considered as independent and the critical chain is the constraint for each single project. In the multiple-project environment, constraints are the heavily loaded resources, which are demanded by multiple projects that restrict the number of executing projects of an organization. The multi-project approach of CCPM identifies the most constrained resource (Drum Resource) which is commonly used across the organization projects and more heavily demanded relative to other resources. If overloaded or not available, this resource is the one most likely to impact on the project duration of all projects. The CCPM staggers the project schedules based upon projects priority in order to remove the drum resource conflict throughout the whole organization, whilst maximizing the throughput of the organization (number of completed projects). This is the objective of the CCPM approach for multi-project environments (Steyn, 2002; Leach, 2005). The staggering of projects across drum resource is also known as *projects synchronization* (Patrick, 2001).

In a multi-project environment projects are interdependent and they share a common pool of resources. Traditionally projects are scheduled as if they are independent. Consequently, it was impossible to foresee how delays in one project impact on the due date of other projects in an organization. As a solution, the Critical-Chain methodology integrates each project schedule into a Drum (pipeline) schedule. The aim of integrating project schedules (pipelining) is to improve the throughput of the organization, also known as completion rate of projects. Hence, the intention of pipelining of projects is not about starting projects as soon as possible, but it is about getting more projects completed.

While scheduling a pipeline of prioritized projects, CCPM identifies a drum resource, which is commonly used across projects, is heavily in demand, and its unavailability constrains the projects' completions. After identification of a drum resource, CCPM staggers the projects considering the drum resource availability combined with individual critical chain project schedules. Critical chain claims in this way the resource conflicts between projects will be decreased which results to increasing the organization's throughput. In the next section the steps for making a CCPM multi-project planning are explained.

3.2.1 Prioritizing of projects

The Critical-Chain multi-project (pipelining) approach begins with projects prioritization in order to avoid multi-tasking for use of the drum resource at project level. This prioritizing can be based on market developments, financial benefits the organization expects to gain by project completion, or any other appropriate organizational measure.

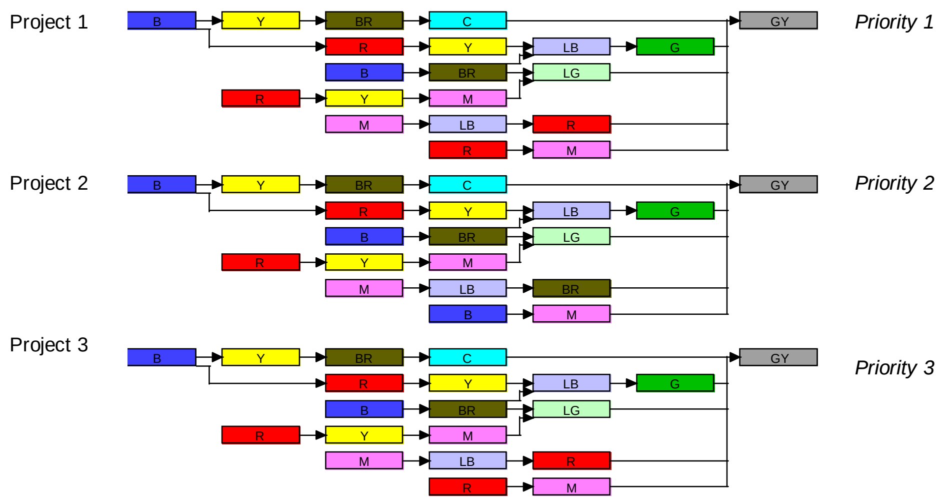


Figure 3.1 Three prioritized project schedules

3.2.2 Staggering (synchronizing) of projects

In a multi-project environment, CCPM removes resource conflicts on the most constrained (bottleneck) resources across all projects of an organization (Patrick, 2001). This happens through staggering (synchronizing) of the projects across the identified drum resource. Figure 3.2 illustrates the staggering of the projects across the Drum Resource (for the tasks with longest duration on the critical chain of each project- in this case, the middle task with a duration of 10 days).

The purpose of identifying the drum resource is to stagger the start of the projects in order to avoid drum resource conflicts between the projects and to reduce overloading for all resources. To achieve the maximum effect of staggering the projects, Leach (2005) suggests that the identifying drum resource should be the resource that controls the largest amount of critical chain time on projects (Leach, 2005, p155).

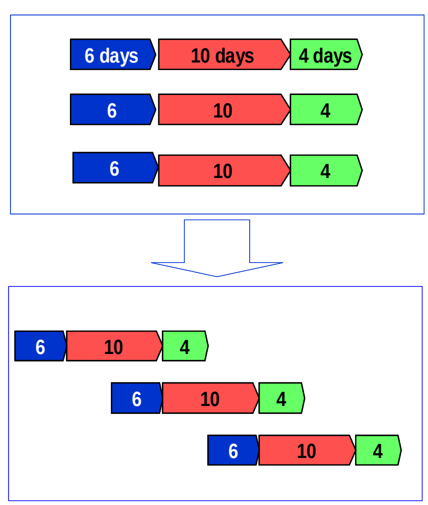


Figure 3.2 Staggering of projects across a (drum) resource

To stagger the organization projects, after prioritization, the drum resource and therefore the constraint on system throughput should be identified. In the Figure 3.1 the red tasks (those are marked with the letter R) are presumed to be the tasks with more heavily demanded resources across the depicted three projects. In the next step, the resource R among the three projects is staggered so that the resource conflicts between the tasks requiring the identified drum resource are removed.

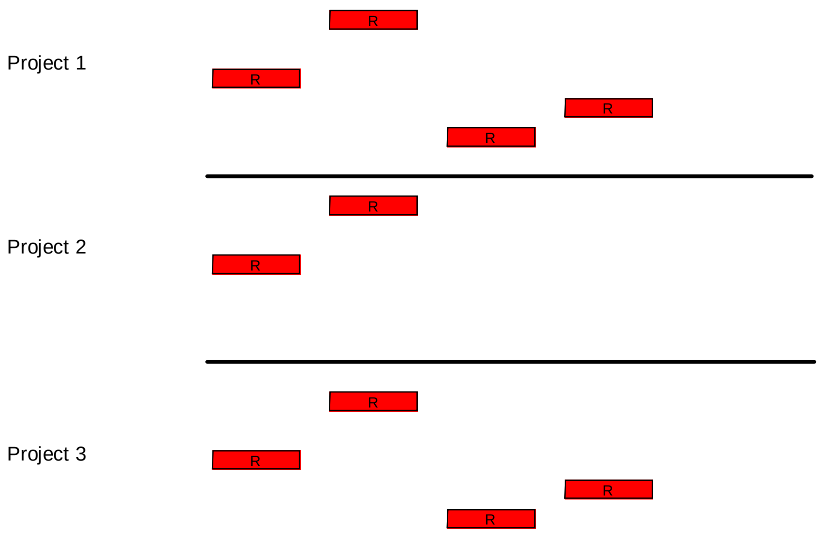


Figure 3.3 Identifying Drum Resource between the projects

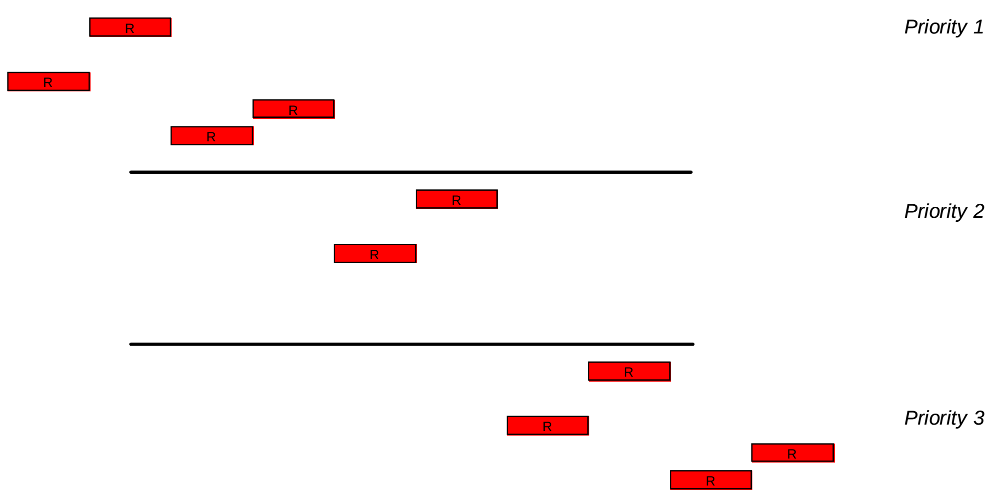


Figure 3.4 staggering across drum resource according to the project prioritization

Staggering the projects in this way reduces resource conflicts for other critical resources, not just the drum resource. This happens especially when the projects are identical or similar (Figure 3.4). Most multi-project environments, however, do not have similar projects. Hence synchronizing of projects to the drum resource eliminates only the drum resource conflicts (Leach, 2005, p151). The remaining conflicts between other resources are absorbed through inserting of so-called Capacity Buffer (see next Section) between the projects (Patrick, 1998).

Kendall (2003) pointed out that staggering projects to the availability of the drum resource e also reduces the effect of bad-multi-tasking between the projects and improves the progress of projects. This also helps to increase the predictability in each project outcome and to increase the effectiveness of critical resources. A shorter individual project cycle time and an increase of the number of projects that can be pushed through the system without increasing resources result from staggering the release of new projects (Kerzner, 2003, ch22).

As mentioned above, with multiple projects CCPM approach does not synchronize all resources across all projects. Instead, it focuses on the most demanded resource shared between multiple organization projects (e.g. department, facilities, and equipment). If increasing the capacity of such most demanded resource is not possible, then it is necessary to protect and monitor workloads. CCPM synchronizes the projects using the drum resource only, and leaves handling the other resource demand fluctuations in the drum (pipeline) schedule to the project and capacity buffers.

CCPM first solves the resource constrained project scheduling problem (RCPSP) for each individual project at the multi-project organization with a simplified approach in order to ensure that there is enough time in each project base line schedule according to the available resource capacities. The CCPM Multi-project approach does not suggest solving RCPSP across all organization projects because during the execution the individual project schedules are still subject to uncertainties, which lead to new resource conflicts among projects and schedule disruptions. The RCPSP is a deterministic technique and the task durations are probabilistic, i.e. exhibit substantial variation. Instead, by staggering each single project in multi-project environment and placing a strategic capacity buffer between projects, CCPM addresses the recourse contention between projects and captures the variability in one project that impacts the delivery commitment of another subsequent project of organization (Leach, 2005).

3.2.3 Inserting of Drum and Capacity Buffer

In Critical-chain multi-project approach, two additional buffers are used to ensure the availability of identified drum resource in the multi-project environment. A capacity buffer is placed between the projects in the pipeline schedule to link the use of the drum resource. While a drum buffer is placed in each project schedule prior to the task demanding the drum resource, that requires starting a preceding task, which does not demand the drum resource earlier so that the drum resource does not need to wait for a preceding task to be accomplished.

The drum buffer exploits the early availability of the drum resource by assuring that all of the preceding tasks are complete if the drum resource completes its task in the predecessor project early. In that respect, the drum buffer operates as a feeding buffer (Figure 3.5). It can be sized as a feeding buffer for the chain of tasks that precede it.

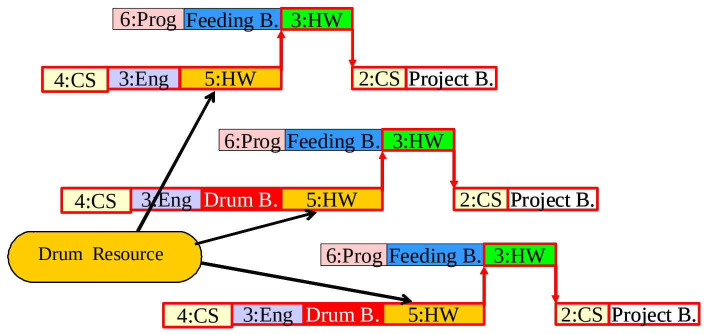


Figure 3.5 Drum Buffers in CCPM Multi-project Scheduling (Source: Lechler, 2005)

The Capacity Buffer ensures that the drum resource is available for the subsequent project in the schedule because the staggering may not be enough to buffer one project from variances in the preceding project causing negative effects to the promised due date of the succeeding project (Figure 3.6). Leach recommends that the capacity buffer should be sized in the range of 25% to 30% of the constraint-resource capacity but does not defend his recommendation (Leach, 2005, p159).

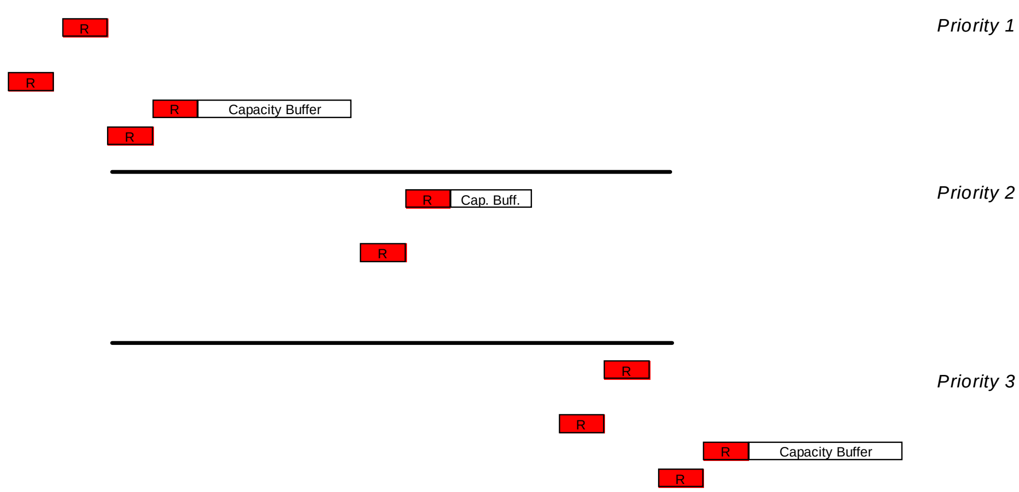


Figure 3.6 Capacity Buffer in Multi-project scheduling

As shown in Figure 3.7, a capacity buffer represents a possible time lag between the completion of work by the drum resource *R* on one project and the beginning of work by the same bottleneck resource on the succeeding project.

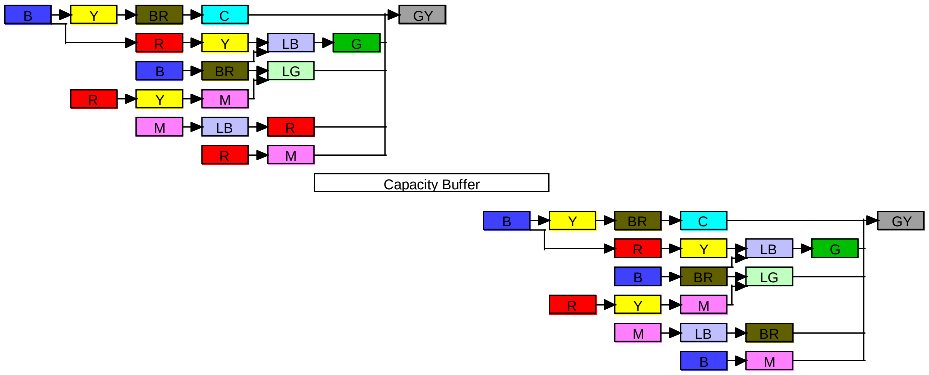


Figure 3.7 Inserting of Capacity Buffer between two projects

3.2.4 Task Priorities between multiple projects in the pipeline scheduling

In traditional project management, the multi-project organizations often tend to launch projects as soon as possible, concurrently with other projects within the organization without sufficient regard to the capacity of the organization. Consequently, conflicting priorities occur among various projects and also the individual projects may claim priority over the use of shared resources to their own advantage. This may result in multi-tasking if the critical resources are shared by multiple projects at the same time in order to move various projects along. When multitasking is the result, the work of task performers may be constantly interrupted by the work being performed on other project, and thus all projects may have to be delayed.

The CCPM Multi-project approach assigns critical resources on critical-chain tasks of the project with a high rate of buffer consumption. If during the execution a critical resource it is required to work simultaneously on several tasks of different projects, CCPM prioritizes the task of the project that is in the greatest risk of missing its committed date, as measured by the project buffer (consumption) consumption rate. According to the CCPM rules, working concurrently on the tasks (by a critical resource) that belong to different projects is not allowed (Raz et al., 2004).

Task priorities and resource allocations are based on the project priorities and the criticality of a project task. In multi-project environment, for each individual project CCPM uses the proportional ratio of critical chain percent complete to percentage of project buffer consumption to set a priority for assigning a critical resource between the concurrent tasks of multiple projects. This proportion is also called *Performance Index* (or Flow Index). Among multiple projects, tasks that lie on the chain with a low performance index got a high priority. This provides that buffers are optimally used, and also reduces pressure to multi-task between projects. For the calculations of chain percent complete and buffer consumption rate see Section 2.3.6.

- Calculating Performance Index for an individual project in multi-project environment:

$$\textit{Performance Index} = \%\ \textit{Chain complete} \,/\, \%\ \textit{Buffer consumption}$$

Example:

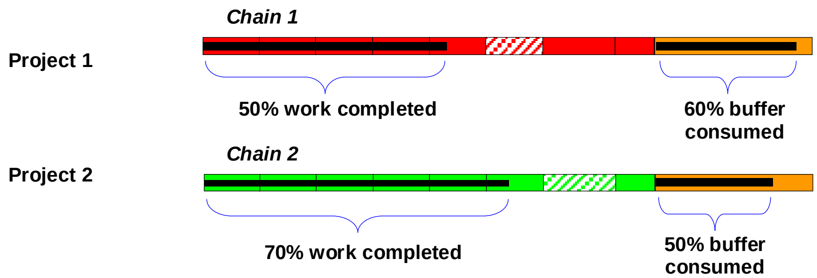


Figure 3.8 CCPM sets the task priorities based on relative impact on buffers of different chains (Source: Realization Technologies Inc.)

In Figure 3.8, in Project 1 the amount of buffers consumption is 60% while the amount of completed chain is 50%. The Performance Index here is defined as the proportion of Chain Percent Complete to Buffer Consumption Rate. The Performance Index for chain 1 then is 0.5/0.6 = 0.83. In the same way the Performance Index can be calculated for the Project 2 as 0.7/0.5 = 1.4. According to the CCPM rules, the hatched task in the Chain 1 of Project 1 has the higher priority because it has a lower Performance Index.

3.2.5 Buffer Management in Multi-Project Critical-Chain

CCPM suggests that the combination of the buffer management monitoring technique and the synchronization of project launches improve the throughput of the organization. Buffer management is the key to tracking project performance in both Critical-chain single or multi-project approach. At multi-project level, buffer management compares the project buffer consumption of each single project to the progress on the critical chain that project buffer protects. The organization projects are aggregated into a portfolio or system level view of the status of all projects in the organization. This is shown in the Figure 3.9:

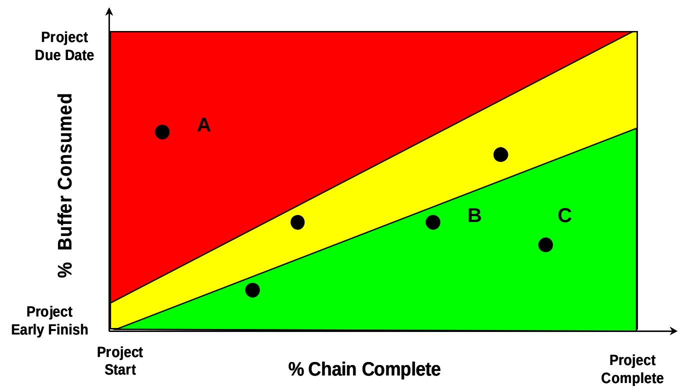


Figure 3.9 Fever Chart reporting schedule progress on a portfolio of projects

In this example, the projects portfolio has good status. Project A is delayed but projects B and C are far enough along and in the green zone so that resources of these projects can be swapped to work on the critical chain tasks of project A in order to get it back on track. In this way buffer management changes the tracking of complex product development environment into simple and effective decision making tool to mange product development process in real-time. Also it provides the focus for determining priority of assigning resources to tasks across all projects.

3.2.6 Inserting of New Projects in the Pipeline (drum) schedule

In a multi-project environment like (new) products development projects, besides the running projects of the organization, new projects may be considered to be launched. It is then required that all projects again are prioritized and the status of all projects to be considered. Therefore, it is first needed to prioritize the upcoming new projects among all the existing projects. If the new project has a lower priority than the current projects, the schedule then determines the start time for the project by taking the priority of other projects and the availability of the drum resource into account. If the new project has a higher priority than a number of running projects, this may lead to interruption or suspending of these projects. The pipelining of projects to the capacity of most demanded resource of the organization (drum resource) helps to decide when and how it is possible to launch additional projects and enables the management to analyze how a new project will impact the project pipeline.

3.3 Critiques of the CCPM Pipelining approach

The critical argument for buffer sizing in the Critical Chain single-project approach also holds for the CCPM multi-project prescription, i.e. for sizing Capacity and drum buffer. Raz et al. (2004) state "CCPM deals with a multi-project environment by staggering the projects around the most loaded constraining resource. In principle, at any given point in time there could be several constraining resources, each leading to a different schedule. The premise that there is a single constraining (bottleneck) resource seems more applicable to the stable manufacturing and operations environment than the most project environment". There is, however, no guarantee that the CCPM pipeline schedule derived from the Critical-Chain schedule of individual projects remains unchanged when the project portfolio changes dynamically or when a new project should be inserted in the pipeline schedule. Hence the need for re-pipelining is inevitable, which it requires gaining agreements gain throughout the organization, while there is no warranty that through pipelining the throughput increases. The other criticism is that by simplistic staggering of projects the resource conflicts between non-critical resources are not directly addressed. This could lead to overloading and pressure to multitasking of other resources.

3.4 Summary

In traditional project scheduling, each project is often managed independently of all other projects. Shared resources force dependencies between single projects, and each project's schedule seldom accounts for these dependencies. Consequently in traditional ways of scheduling it is impossible foresee how delays in one project impact other projects of the executing organization.

In multi-project environments, critical chain approach connects all projects in a multi-project environment together in a pipeline by primarily identifying the most constraining resource within the multi-project organization. This constraint is a common resource between projects that is heavily loaded and forms a bottleneck (drum) resource for the release of new projects. It determines the pace of all projects within the portfolio. Each project is then scheduled across the drum resource in a manner to maximize the usable capacity of the drum resource. In other words, the start of new projects is staggered so that new projects reach the drum resource sequentially, based on priority, due-date, etc. This is different from traditional project scheduling, where each new project is released to operations to start the work as soon as possible, independent of all other projects in the pipeline. The CCPM suggests that the staggering of projects through the resource constraint help make the interdependency visible for all projects in the pipeline, and it results in the reduction of bad multi-tasking and of the number of active projects in the entire organization at a given point in time. It also minimizes the resource conflicts between the projects which lead to projects that are completed in shorter durations or that more projects can be delivered in the same time period with the same resources (increasing throughput). Also, the impact of any new projects can be determined quickly and it helps to decision making process for the launching of new projects inside an organization.

Summarized the Critical-Chain multi-project is about performing three sequenced rules:

1. Buffering of single projects (to manage uncertainty)

    - Setting aggressive task duration estimates
    - Scheduling tasks as late as possible from right to left
    - Developing Precedence and resource dependent baseline schedule by solving RCPSP
    - Identifying critical chain of each project

2. Pipelining of single projects (to increase throughput)

    - Prioritization of projects to meet drum resource demand
    - Staggering projects across drum resource capacity instead of starting as soon as possible
    - Enables the management to analyze how launching a new project will impact the pipeline schedule

3. Buffer Management (to prioritize tasks)

    - Monitors the tasks that are most effecting the system throughput and projects due date
    - Provides task priorities within the entire portfolio of projects with a common pool of resources
    - Provides early warnings to allow the management and the project team to prepare recovery actions at an earlier stage of project execution.

4 CCPM application at Bosch CCTV Eindhoven

4.1 Introduction

The purpose of this chapter is to address the second research question:

- Are there differences between the theoretical Critical Chain rules and principles and the application of CCPM at the Bosch CCTV Eindhoven development department?

In this chapter we describe the characteristics of software development projects. Further, we survey the CCPM application for (software) product development projects at Bosch CCTV, and finally we provide an overview of perceived strengths and deficiencies regarding the CCPM rules and principles as explained in Chapters 2 and 3. We interviewed different CCPM practitioners at the CCTV development site. In this study we distinguished between application or deployment of Critical-Chain concepts and CCPM implementation. With Critical-Chain application, we mean running projects according to Critical-Chain rules and principles at project level without regarding to how the rest of the organization runs its projects. CCPM implementation means making it part of the daily activity of project's stakeholders in the organization from executives to project team.

4.2 Nature of software development projects

In this section we describe the general characteristics of software projects, and the reasons that make scheduling and executing these projects more difficult and uncertain.

Invisible and intangible developing process: Software products are generally as complex as the hardware on which it runs. Hence, hardware delivery usually depends on successfully developing software. In comparison, software is an intellectual product as opposed to a physical or hardware product. Intellectual products are intangible and largely invisible. This makes software development projects more challenging because both the software and the procedures for creating the software are not routine work.

Non-routine tasks: Also, the result of a software project is always a unique product and consequently development tasks are mostly non-routine. Software development has few routine tasks and includes a significant design component which requires creativity and innovation. Together these characteristics make software projects more uncertain and less predictable than hardware projects.

Human-intensive tasks: The software development processes are human intensive and the duration of a task in software projects depends mostly on the skill of the human-resource assigned to the task as well as the resource learning rate. Hence, if a task is executed by an alternative task performer, this may result the different expected task durations. Therefore, a managing approach proposed for software development projects should incorporate resource assignment features in order to cope with the uncertainties caused by the human-intensive processes.

Customisation: If a project involves the custom-development and customisation of software components based upon the needs of a customer, it is completely unique and contains more uncertainty and risk compared to repetitive, standard, tangible or physical projects. The reason is that software development projects incorporate the acquisition of intangible data (requirements of the customer which are often imprecise) as well as the involvement of a team of task performers whose skills cannot be measured with certainty. This all makes the software development process more complex and uncertain (Özdamar and Alanya, 2001).

Multi-project environment: Software projects are usually done in multi-project environments. In reality, the development projects are not carried out in a single project environment. In software development, projects with a great volume of scope and lead times are decomposed into manageable number of so-called sub-projects and assigned to a shared resource pool. This is a multi-project environment in which projects compete to use resources from a common resources pool within the organization. These projects are often complex regarding to the interdependencies in resources, sizes, priorities and progress levels (Lee and Miller, 2004).

Uncertainty is inherent and inevitable in software development processes and products (Hadar et al., 1996). The Critical-Chain approach claims to deal with the uncertainties mentioned above and provides an approach in order to capture the uncertainties incorporated in the project duration estimates. Critical chain scheduling differs from the traditional project scheduling methods in how uncertainty is managed, by aggregating safety times of individual tasks and using them in the form of protecting buffers in order to absorb contingencies and uncertainties during the projects execution. CCPM also deals with the negative behaviour of human-resources and provides an approach for multi-project domain wherein multiple projects share a common resource pool in the organization. These all indicate the Critical chain methodology as an alternative for planning and managing of software projects.

4.3 Product development at Bosch CCTV

The Bosch CCTV Eindhoven (until 2002 Philips Communication, Security and Imaging) develops products for security means namely CCTV cameras and Digital Video Recorders (DVR). Cameras and recorders consist of Hardware (HW) or technical parts and embedded Software (SW) or functional parts as their main components. The structure of the development organization is also concentrated around the main two hardware and software development activities (Figure 4.1).

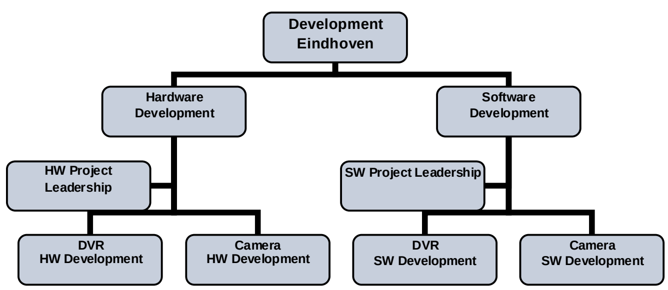


Figure 4.1 Project organization structure of Bosch CCTV

Figure 4.2 shows the phases of the Product Realization Process (PRP) within CCTV Business Unit, each with milestones marking the transition from a phase to the next. New product development spans the period from initial product definition to mass production phase. These phases are almost identical for different types of products. The development phases of products are project based, which means that the phases and the activities within each phase are managed as a project.

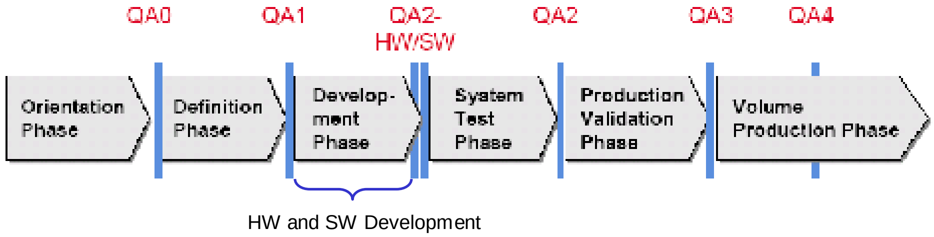


Figure 4.2 New Product Development Phases and Milestones

The period between the QA1 and QA2 milestones forms the more practical and time consuming stage of the whole development process. The development phases for software (SW) and hardware (HW) are organized and managed in parallel. After the accomplishment of the hardware and software development phases, the developed hardware and software components are integrated. Depending on the market situation and adjustments in (product) development targets, it may be decided to develop further hardware, software or both components of a (new) product. Figure 4.3 illustrates the parallel process of HW and SW development.

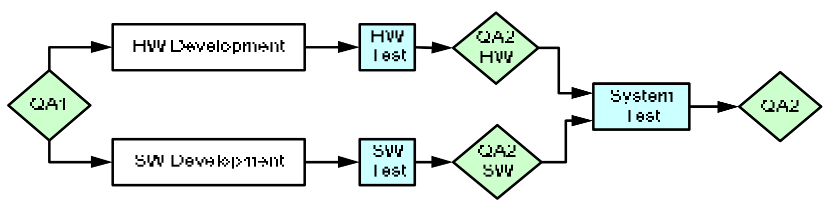


Figure 4.3 HW and SW Development Phases (QA1-QA2)

In today's digital market, new security products constantly possess more functions. These product functions are facilitated through software embedded within it. The more functions a product is planned to have the more complex the software component needs to be. Hence, in today's increasingly digital world, software development has become a critical component of product development. Software projects are uncertain by nature, and this causes project delays. Regarding the Critical-Chain approach and its guidelines in order to deal with incorporated uncertainty in the project schedule, compared to traditional project scheduling techniques, it can be an alternative for project scheduling and monitoring of (software) development projects.

4.4 CCPM application at Bosch CCTV

The Bosch CCTV development department also experiences the complexities mentioned in Section 4.2 in its software development projects. In order to cope with the high levels of uncertainty in the development phases of the software components and to increase the reliability of the project schedules and predictability of project promises, Bosch CCTV decided to utilize CCPM system in its development departments. The reasons behind this decision were the promising benefits of the Critical-Chain application (Leach, 2005).

To gain an overview about the current situation and also to get information about the findings and experiences of Bosch CCTV about the CCPM application, we interviewed 13 CCPM practitioners (1 senior manager, 2 Project Managers, 4 Task Managers and 6 Task Performers) at the development department who were responsible for the execution of a new software project. This software project was also launched as a CCPM pilot project in order to embark on the CCPM principles and work methods. To support the CCPM deployment, Bosch CCTV purchased the Concerto tool. Concerto is a web-enabled CCPM software package which uses the popular Microsoft Project planning software package within its framework, and automates the CCPM processes described in Chapters 2 and 3 in both single and multi-project applications. The pilot project network was fed into the Concerto tool, and then the project buffer and feeding buffers were calculated automatically.

The interviewees had been taught earlier about the CCPM theoretical principles and rules and had attended different CCPM training workshops. In the interview we asked the project staff about their findings and experiences with the currently implemented CCPM at Bosch CCTV. In this section we first sum up the subjects (themes) of the questions in the interview. Next, for each question's theme, we compare the experiences and findings of the CCPM practitioners at the Development Department with the principles and concepts of the CCPM approach. Finally, a summary will be given. In Chapter 5 we pay attention to the CCPM implementation aspects in an organization and areas for improvement of the CCPM application at the Bosch CCTV Division, and give some recommendations in order to improve them.

The interview consisted of different questions about the reasons of implementation of CCPM system at CCTV, objectives achieved through implementation of it, the weaknesses and strengths of CCPM principles in practice, application of CCPM in software development projects, the positive aspects/lessons learned from CCPM deployment at CCTV BU, and the negative aspects and observed differences in CPM usage at the CCTV department regarding CCPM principles. Below an overview of these question and answers to each is given.

4.4.1 Reasons of the CCPM adoption

From the interviews the following were identified as the major reasons for the CCPM adoption at the development division:

To prevent project delays; project delays and schedule overruns are named as the main reason for deciding to apply CCPM at Bosch CCTV.

To increase visibility or transparency; also attempts to get more visibility in project status or transparency are mentioned as important motivations.

To increase predictability (schedule reliability); predictability or schedule reliability is another reason embarking on the CCPM approach. Using protecting buffers and monitoring of them helps to operate proactively and to prepare recovery actions to cope with possible delays.

To increase throughput; increasing the projects flow i.e. completing more projects in the organization with the same resources

Many examples of successful application of the Critical-Chain methodology in the literature indicate that these objectives are achievable (Leach, 2005). One of proven benefits of Critical-Chain is the simple and consistent view of the project status. The measurement of the project buffer consumption rate provides an easy and simple method to monitor and report the project status. Traditional project monitoring methods do not provide an obvious or early indication that a project schedule is going to overrun. The CCPM approach provides sufficient early warning signals. The early warnings create opportunities to allow the management and the project team to prepare recovery actions at an earlier stage of project execution.

4.4.2 The objectives achieved through CCPM deployment

More insight and visibility; the answers that are given about the achieved objectives indicate that through CCPM deployment, more insight into the status of the current development projects is given. Also the predictability in terms of promised project due date to both internal and external customers is improved, which means that it is possible to see what the actual status is of the project execution and when the project with the current execution trend will be finished. This gives opportunities to the project organization to operate proactively and prepare recovery actions when there are indications that the project due date cannot be met.

Shorter delays; in the opinion of the interviewees, despite planning and managing of projects according to the CCPM approach, schedule delays still happen. They have the impression however that by using CCPM the delays are shorter than in the past. They named the complexity of projects, unforeseen scope requirements or changes, and inability to sound detailed planning because of technological uncertainties as the major causes of the recurring project delays.

Increasing predictability; another achieved objective is that by means of the early warning operation system through buffer management system in CCPM, the tracking and removing of conflicts and problems can be done in an earlier stage of project execution. This provides more predictability for project managers, project owners and suppliers and assists the decision making process by early warning system. CCPM anticipates when the project with the current rate of buffer consumption will be finished.

Improving throughput; the improving of throughput (doing more projects with the same resource pool) at Bosch CCTV actually was not achieved. Unfortunately at the time of doing this research, the utilization of CCPM multi-project approach at the Bosch CCTV development department was still in the experimental stage and consequently there were no findings or outcomes that confirm the promised improvements of Critical-Chain multi-project approach as mentioned in Chapter 3.

4.4.3 The strengths of CCPM approach perceived in practice

a) Transparency and predictability; CCTV observed that through CCPM deployment more visibility in the project status – transparency & predictability is achieved. The supporting tool of this CCPM feature is the buffer analysis. The buffer trend (fever) chart reflects the rate of project buffer consumption during the project execution. It provides a good insight into the project status and predicts when the project with current execution speed can be completed. The buffer consumption rate indicates the speed in which the running-project is progressing.

b) Focus on essentials; the interviewees agreed with the claim that the Critical-Chain approach focuses on essentials (critical tasks/resources) and has a direct impact on project due date. Through buffer management, it is possible to make sound priorities and focus on critical tasks and resources. For the working mechanism of buffer management, see Sections 2.3.6, 3.2.4 and 3.2.5.

c) Early warning mechanism; the buffer management of CCPM acts as an early warning mechanism and it is a useful tracking system. Through monitoring of the buffer consumption rate in the fever chart (snapshot of current project status) it is possible to see the current status of the project and whether the project can be completed on the promised due date. If the amount of buffer consumption is high, this indicates that the project due date with this high trend of buffer consumption is not achievable, and it is time to prepare and to undertake recovery actions to compensate the buffer consumption rate and bring the buffer usage rate to the normal situation.

d) Coping with resources negative behavior; the consciousness about the negative nature of resource behavior helps to increase alertness in order to save time. The task performers that work according to the critical-chain rules on the project tasks are more conscious of the obstructing behavioral factors, e.g. multi-tasking, gold-plating and trying to avoid them as much as possible. However, the complete elimination of these negative behaviors is not always possible in practice (see Section 4.4.4).

4.4.4 The weaknesses of CCPM methodology encountered in practice

During the executing of software (interdependent) mini-projects, project personnel also encounter some drawbacks of CCPM.

a) Sizing of buffers has not a solid basis; Critical-Chain prescribes cutting the task duration estimates by 50% and sizing of buffers about 50% of the length of the chain leading to each buffer. The CCPM claims that an aggregation of task safeties in the form of buffer provides a better protection regarding single task safety times. Using the 50% or Cut and Paste method (see Section 2.3.8) provides a short (or aggressive) buffered baseline-schedule that is shorter than the original un-buffered baseline schedule. The Bosch CCTV handles this method differently than its original form. They first increase the task estimates by 25% to account for the possible delays as a result of the absence or illness of the task performers which are not easily replaced with another task performer. In the next step the task duration are cut to 66% and the rest (34%) is put into buffers. Consequently the resulted aggressive buffered base-line schedule is longer than the 50% cut and paste method but it almost has the same duration of the original un-buffered baseline schedule. Bosch CCTV targeted initially the protecting of projects due date rather than shortening the duration of project (schedule) at once in order to get experienced firstly with the practicing the CCPM rules and principles, and to increase the reliability and predictability of scheduled project due dates toward its sales office. In spite of simplicity of the Cut and Paste method for sizing of buffers, this method is however dissuaded for the product development projects, because of providing unnecessary and uncompetitive large buffers (see Section 2.3.8). Despite this we encountered the inserting project buffers were totally (unrecoverable) depleted during the pilot project execution, because the assigning buffers were not sufficient to cope with the significant scope changes and appearance of unforeseen new tasks as consequence of technological uncertainties and risks which are known as special causes of variation. This indicates the need to bigger buffer sized in order to capture different kind of uncertainties in the product development environment,

b) Re-scheduling the CCPM schedule is inevitable; in product development projects, it is not often possible to foresee the future tasks with consistent detail over the entire period of the project. To address this issue, the development projects are mostly scheduled by the so called Rolling Wave Project Planning (RWPP) method. The Rolling Wave scheduling provides a project plan with detailed short term estimates right from the beginning and saves extra efforts for (an inaccurate) long term planning which is open to changes and uncertainties, and will not provide stable due dates. The focus is mainly on short term tasks and the future tasks are broken into detail when their executing time is coming up. This method phases the project plan with higher level of detail and lower level of uncertainty associated with the near-term and less detail and more uncertainty for the later phases of the project. Also despite the use of this approach the technological uncertainties are still incorporated in the future tasks of the project schedule because at the beginning phases of project making of accurate estimations are not possible. Therefore, it is difficult to define the required size of project buffer to protect the project schedule against technical uncertainties, risks and common causes of variation. This necessitates periodically re-scheduling of the project schedule and re-sizing of the project buffer when significant scope changes occur in order to get an update number of remaining tasks and to focus on the new critical tasks as the project further develops.

c) CCPM is not comprehensive; according to the opinion of the interviewees using only CCPM is not significantly more effective than traditional project management methods for planning and managing of product development projects with a poorly defined scope, irregular scope changes, or gross estimation of needed level of effort as a result of technological uncertainty in the (product) development projects. The present technological uncertainty in the (product) development projects may have an influence on the schedule, cost and success of the project. The time buffers in the CCPM approach are primarily created in order to absorb the common cause variation in the known tasks of the projects. The CCTV managers refer to quickly landing in the red zone of the fever chart as a consequence of uncertainties in the scope of product development. This led gradually to dissatisfaction of the tasks performers when despite devoting sufficient effort on their executing tasks were still in the red zone of the CCPM fever chart. Leach (2005) points out that potential causes of risks or special cause of project variation should be addressed by the conventional risk management method and state “risk monitoring, prevention, and mitigation should be part of the project plan”. Also, Kendall et al. (2001) support this recommendation and state, “Risk management in planning is performed to identify, analyze and respond to project risks. It results in an understanding of how much time should be allocated to the project buffers”. Therefore it is required, except the CCPM deployment, for high uncertain environments, a risk management process in order to cope with the special cause of variation, e.g. technological uncertainties and significant scope changes, is included in the project planning for the estimating of tasks and sizing of buffers.

d) CCPM procedures are intensive and time consuming; as mentioned above the interviewees consisted of people from different functions and responsibilities in the project mission, e.g. project managers, task performers who have different expectations and points of view about the implemented CCPM system. Hence, it is possible that some of the observed drawbacks by people from one functional position, e.g. task performers (resources) are not visible for the other group, e.g. project managers. For example, from the task performer’s points of view, the CCPM tracking and updating procedures are intensive and time consuming, e.g. attending the daily meeting with a task manager. In this meeting, issues like reviewing of the remaining time they need, reviewing of the check list, discussion about possible interruptions and issue resolution are addressed.

e) Avoiding negative behaviors is not always feasible; from the project and task managers’ point of view, the CCPM behavioral modification (e.g. preventing bad multi- tasking) is not always feasible in practice. For example, the resources with a supporting role, e.g. software architects are mostly involved in different tasks. During the software project execution, these resources simultaneously support performing of different tasks and assist the task performers. So they cannot wait until their current task completely finishes, otherwise the dependent task performers of other tasks have to wait for them and consequently they cannot meet the estimated task duration. Thus the elimination of bad multi-tasking is not always feasible.

f) Placing of resource constrained tasks disregarding required execution sequence in the schedule; in the CCPM (resource constrained) scheduling, tasks that demand the resources from the same type, in case of limited availability of resources, are randomly leveled or positioned in the schedule where enough resources of the same type are available. This may not be appropriate in practice because during the project execution some tasks are not only technically dependent, but they may be preferred to perform by the same resource(s), i.e. the out put of the precedence tasks is the input for the same resource(s). If the resources are overloaded, the Critical-Chain scheduling, however, levels the tasks demanding these resources unnoticed and place them in other un-overloaded places in the project schedule where enough resource of the same type are available. This leads substitution of resource(s) while performing of the next tasks by the same resource is desired or assigning them to the new tasks with inputs from the precedence task from the other recourses. For example, when executing a software design project, for each component, tasks like Specification, Design, and Coding are desired to be performed sequentially by the same task performer of one resource type. For example, for three individual components X, Y, and Z the Critical-Chain (resource-constrained) scheduling may schedule the Specification task of component Y next to Specification task of component X instead of Design task of component X, and etc. This is not appropriate in practice because the same task performer who is executed the Specification task of component X is able to directly start the Design task of component X and does not need extra switching time because the hand-off (the output of the predecessor task) is already provided by the same resource. Otherwise the total completion time of required tasks for component X takes longer than when they executed consecutively by the same task performer.

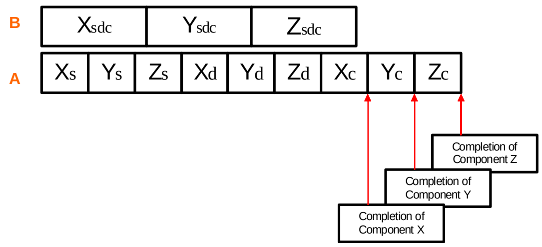


Figure 4.4 Longer completion time when the tasks (specification, design and coding) for each component (X, Y and Z) are executed by different task performers or when they are executed separately

g) Interchangeability of resources; in product development projects usually human-resources like knowledge workers are involved. This makes the appropriate utilization of the CCPM resource assignment feature difficult. As mentioned earlier CCPM dynamically assigns the resources based on task priorities that are defined by the proportion of buffer consumption to percent complete of chain feeding it known as Performance Index in an individual projects. In a multi-project environment, in case of equal Performance Index among projects, the project with the higher priority is eligible for assignment of the critical resource. But in practice it is not always feasible that resources after completing work on their tasks are swapped between the projects or even between the tasks of a project. Once the project has started, this flexibility (resource interchangeability) starts to decrease, slowly at first and increasingly as the project moves on. It is moderately difficult to swap resources if a chain of tasks is not yet started or no affinity to the task has been developed. The further the chain progresses, the harder resource swapping becomes. It is very difficult to swap resources in the middle of a chain (for example in software developing process, analysis and design are done, and then to swap somebody else to write the actual code). This is because the original resources have developed a knowledge pool that would need to be transferred to the new resource (task performer). This requires extra time or revision of the task duration that almost negates any time benefit of resource swapping. However, the resource interchangeability in development projects wherein knowledge workers are involved is not impossible by any means. It is inevitable when the primary resources have suddenly become unavailable (illness, left the company), but it is difficult. The answers of CCPM practitioners at the CCTV development department confirm the above mentioned drawback.

h) Time is the main focus; CCPM focuses on the project schedule as a main dimension or aspect of project management and it provides no guidelines to the other two important project constraining dimensions, i.e. budget and quality. Therefore in the Critical-Chain approach, time is the main focus. The critical chain approach is in fact a scheduling and monitoring technique. It does not directly provide solutions for improving other project management objectives like cost and quality. The CCPM targets to shorten projects cycle time with the same amount of resources. However the time savings during the project execution or speeding up the project progress may negatively influence the delivered quality and budget of project. Most of the task performers (software engineers) within Bosch CCTV, however, agreed with this claim because they believe that the time savings sometimes are only achievable at the cost of quality or budget. Also focus on short-term accomplishments may result in neglecting other important work and losing sight of project goals and being reactive instead of proactive toward foreseeable events and contingencies.

4.5 Summary and Conclusion

Although the Critical-Chain approach gives more insight into the project status, the experiences at Bosch CCTV Eindhoven indicate that it is unable to catch the all uncertainties embedded in software (product) development projects. The inserted time buffers in the CCPM project schedule are only aimed to protect the project from the common cause variation in the known or foreseen individual tasks of the projects. Incomplete scope definition, appearance of new tasks are examples of special cause variations (or technological uncertainties) that should be addressed via proactive risk management techniques, or considering reactive risk mitigation plans by additional prepared resources when the risks strikes. Hence, it is not realistic to count on the inserted buffers to make up for technological risks and drastic scope changes.

Development projects are uncertain by definition. Hence, it is required to be prepared for the expecting schedule delays and large amount of buffer consumption. When the project buffer is totally consumed, to bring the project back on track, recovery actions should be taken, otherwise the remaining project tasks should be rescheduled and a new commitment date should be considered. However, in the highly uncertain environment of (product) development projects, often big changes or adjustments in project estimates and schedules may occur. Therefore, to create and maintain a realistic project schedule, it is required to frequently re-schedule the project network and to negotiate the new project due dates with the project stakeholders and external parties.

The Critical-Chain methodology suggests that during the project execution, for recovering the detected tasks with the red buffer status, it is possible to pull resources from the non-critical tasks, i.e. with the green buffer status, and dispatch them to the tasks with the red status. In the product development project where mostly high skilled people and the so-called knowledge workers are engaged, the CCPM prescription for interchanging resources is difficult to apply. In practice, as the project progresses, the resource interchangeability decreases. It is then difficult to swap resources when a chain of tasks is started and affinity to the tasks has been developed. The further the chain process, the harder the resource swapping becomes. This is because the original resource has developed a knowledge pool that would need to be transferred to the new resource which takes more time than what is remaining or estimated in the original task. In particular circumstances when resources have suddenly become unavailable, i.e. illness or leaving the company, the need to swap resources is inevitable.

In spite of the perceived deficiencies of CCPM methodology in the practice of (software) development projects, Bosch CCTV is positive about the CCPM adoption at its development division because of the observed benefits of the CCPM application, i.e. visibility of project status, simple project monitoring, predictability in setting of project due date, and visualizing the negative human resource behaviors in the scheduling and managing of the project tasks.

In the next chapter, we explore the implementation aspects of CCPM at the organizational level and how these are addressed at the CCTV development department.

5 CCPM implementation aspects and the way these are addressed at Bosch CCTV

In this chapter the CCPM implementation aspects at the organization level are discussed. This chapter aims to find an answer to the third research question:

- What aspects are required for deploying CCPM in an organization? How far these are attained at the CCTV development department?

In order to answer the third research question, we first provide an overview of important aspects for the CCPM implementation in an organization. Next, we study the CCPM implementation at the Bosch CCTV department. To this end, we interviewed some of the project personnel at Bosch CCTV about their experiences and findings from implementing the CCPM system at the development department.

5.1 CCPM implementation aspects

When reviewing the Critical-Chain theory as described in Chapters 2 and 3, we indicated that the CCPM theory is easy to understand, but it is challenging to implement, because implementing CCPM requires a new way of thinking about planning and managing of projects, and changes in mindset and behavior, i.e. the culture of the entire project organization. Also, Devine (2004) states “while the CCPM concepts and rules are easy to understand, the implementation of CCPM is more difficult due to the culture change required throughout the organization”. Therefore exploring and identifying the required changes or adjustments for implementing CCPM and addressing them are

Meanwhile many organizations successfully tried to implement CCPM methodology for managing and monitoring their projects, but of course not all of them had success and some had to give up (Leach, 2005 p187). The statistics of 150 CCPM implementation attempts indicate that only about one third of CCPM implementations were successful. Also despite the initial successes, about 15% of these implementations failed to be maintained (Gupta, 2005).

5.1.1 Critical success factors for the CCPM implementation

The necessary changes and requirements below are typical reasons that make the implementation of CCPM approach a challenge for many organizations. These requiring changes form the important aspects for implementing CCPM within an organization, and are explored through experiences and lessons learned form many real-world implementation attempts in the different companies that shared or published in the CCPM literature.

Obtaining the organization’s endorsement; support and involvement of the senior management and the project staff, and full understanding of the CCPM implication by the project stakeholders are one of important success factors for the CCPM implementation in an organization, because pursuing the Critical-Chain rules in practice needs a paradigm shift and requires a change of mind-set, behavior, culture and rules within the organization. Experiences from the different CCPM applications show that changing the work habits of the task resources (from task switching to relay racer) is easy if management supports the change (Simpson, 1999; Leach, 2005). Organizations still using an older paradigm or a traditional project management methodology show especially difficulties to implement a paradigm shift and break the old habits. For example, for a software development organization it is not easy to convince the software engineers to do things very differently from what they have been doing for a long time, even if it is logical and correct, the need is great and its benefits are big. Hence, the implementation of CCPM should be carefully organized to move people from possible resistances to change, which can impede implementing CCPM and exploitation of its potential benefits (Zultner, 1998). Therefore, obtaining endorsement of project stakeholders is an important success factor for implementing CCPM in an organization.

Changing old habits and behavior; implementing the required changes for the CCPM implementation in practice, i.e. working with aggressive task duration estimations, starting tasks as soon as preceding tasks are done, finishing open tasks (WIP-executing tasks waiting to be completed) as quickly as possible, avoiding negative behaviors, e.g. bad multi-tasking, student syndrome, reporting early finishes of WIP tasks by task performers (resources) are seen as critical success factors for effective implementation (Leach, 2005; Devine, 2004).

Clear defining and assigning roles and responsibilities; besides the CCPM scheduling rules, defining roles, determining responsibilities and providing internal procedures are influencing factors on the project due date time within the executing company. Task management is one of key functions in the CCPM deployment within an organization (Leach, 2005). Task managers play an important role in pursuing and maintaining the CCPM rules and principles through coaching of task performers in the executing of their tasks regarding the Critical-Chain prescriptions (see Section 2.3.7). Although there are multiple software tools (packages) available for supporting the CCPM processes, however, these CCPM (software) tools are not an answer in itself without fundamental changes or adjustments in the roles, behavior, culture and rules of the organization (Lechler et al., 2005b). For effective CCPM utilization, it is therefore required to define whom are responsible for planning, updating and completing planned tasks and monitoring and protecting project schedules through using the buffers analysis, e.g. defining task priorities and anticipating of possible delays, and identifying and taking recovery actions.

Changing performance-measurement attitude; the CCPM implementation requires adopting a different mentality and attitude when judging those responsible for completing tasks, working according to relay race mentality (see 2.3.4), and providing reassurance for task performers that there are no penalties for delays regarding aggressive duration estimates and micromanagement of task performers (to manage with great or excessive control or attention to details) are avoided. They are asked to report early finishes that contributes to the project buffer protection. Also, instead of asking the completed percent of work, tasks performers are asked frequently to report the remaining duration of running-tasks, interruptions and obstacles in order to smooth out as quickly as possible execution problems (Patrick, 2001).

Shifting focus from the individual tasks to project due date; CCPM insists on continuous improvement to meet the system's global objective, not the local one. This means in the single-project domain changing the direction from protecting individual tasks (local optimization) to protecting the project as a whole (global optimization) and in the multi-project domain shifting focus from protecting each project to protecting completion rate of multiple projects (throughput). CCPM explicitly emphasizes on shortening of project duration and increasing of throughput as objectives for respectively single and multiple project environments (Lechler et al., 2005).

Communication with the external parties; Submitting of Critical-Chain planning with the external suppliers and third parties; the CCPM project schedules are different than traditional project schedules, because they are shortened (or aggressive), and contain buffers instead of safety times or due dates (milestones) for each task, and non-critical tasks are scheduled as late as possible. Therefore the external parties, suppliers and clients should be informed about these changes in the organization's schedules and procedures when using the CCPM methodology (Lechler et al., 2005b).

Sustaining of the implemented changes; Also, sustaining the implemented changes for the CCPM deployment in the organization is an important factor for successful CCPM implementation. The existence of the CCPM reference points or experts within the organization is an important supporting factor for maintaining of the implemented changes. Also, evaluating of the initial outcomes and addressing improvement areas contribute to sustaining introduced changes and understanding of them. Otherwise the organization reverts to its previous manner of working, i.e. including safeties in each individual task, working with task due dates, etc (Leach, 2005).

5.1.2 Piloting CCPM implementation

The CCPM implementation itself can be seen also as a project. Depending on the specific needs of an organization it can vary in content and scope. The implementation is an incremental process, and its success requires not changing everything at once and depends on the behaviors the organization currently demonstrates. It can be piloted first with one or two projects in order to obtain endorsement of project personnel or applied to all organization projects simultaneously, but this latter approach needs more preparation and coordination efforts and trainings of project personnel throughout the entire organization before beginning the execution. However, each of these approaches has its own advantages and disadvantages.

Conducting a CCPM pilot project shows how the Critical-Chain concepts work in practice and helps to develop a practical understanding and belief in the CCPM methodology. It is intended to get the project team acquainted and skilled using the CCPM rules and to determine which changes within the organization are needed. It helps also to make preparations in order to expand the CCPM application to the rest of the organization, to make the CCPM implementation successful. For the CCPM piloting, a (few) project(s) and a resource group within the organization are required. The advantage of using a pilot project is that it requires less training effort and achieves faster results. In case of successful outcomes of pilot project, it is intended that the deployment of CCPM is further roll out to other projects. The pilot project helps also to create a practical internal capability to exercise and teach the CCPM rules in order to expand a critical chain planning and control system across the organization.

The CCPM pilot project increases also the improvement opportunities for the next CCPM projects by incorporating of experiences and lessons learned from CCPM application on it. In addition, applying CCPM to a single project as the pilot is simple and easy to understand. Also in a multi-project organization with the shared resource pool, it is reasonable to pilot a single project if possible. This requires, however, extra care (Herroelen and Leus, 2002). Alternatively, a direct piloting of the full multi-project solution is not impossible, but it requires more effort and preparation that may make it risky. Also, conducting a CCPM pilot-project requires less initial investment of the organization, and in case of failure, the CCPM implementation can be stopped to save further effort and costs. In the literature there are no specific reasons mentioned that company-wide implementation approach or simultaneous deployment of CCPM to all organization projects is better or worse than an incremental approach. However, depending on the organization and its project types these approaches can have changing priorities.

5.2 The CCPM implementation at Bosch CCTV Eindhoven

Bosch CCTV Eindhoven is the Development Division of advanced digital video security systems, mainly Cameras and Digital Video Recording (DVR) products. Because of market competition these products get more functions, which require more and more embedded software. Hence the development projects for these products became increasingly more complex and uncertain, and raised gradually difficulties for the planning and control of these projects. To cope with the growing complexity and uncertainty in the product development projects, Bosch CCTV decided to embark on the Critical Chain Project Management methodology.

5.2.1 Brief history

Bosch CCTV found out about the CCPM approach through the Atos Origin Co., one of the suppliers of the Development Division which had taken over the KPN software house (formerly Lucent) in 2002. The first isolated trial of Critical-Chain methodology without the CCPM tool support (software package) was done in 2003, the implementation steps were continued as follows: serious orientation in May 2004; involvement of consultants of Realization Co. in August 2004; first (software pilot-project) project schedule in Concerto (CCPM package) in December 2004; start of second phase: CCPM Multi-Project Pipelining in September 2005, further roll-out to projects of Camera portfolio in the Development Division anticipating by then proven success of earlier projects during 2006.

5.2.2 Current situation and achievements

In this section we surveyed how far the mentioned critical success factors and the required changes (Section 5.1.1) are implemented within the CCTV development department.

The CCPM implementation at Bosch CCTV was done through a top-down process and the senior-management support and involvement was the starting point of the CCPM implementation within the development division. According to the CCPM adherents at the Development Division without the senior manager's buy-in and involvement of the CCPM adoption within the Development Division was not possible.

For the training of project personnel, and supporting the CCPM implementation activities within the development department, e.g. understanding CCPM scheduling and tracking techniques, applying the CCPM supporting package, etc. Bosch CCTV decided to involve the Realization firm of consultants. The project personnel at the department experienced the organized workshops and training sessions, a useful way to understand the CCPM principles and rules. The experiences of task performs indicated that through the CCPM behavior modification guidelines task performers became more conscious about the negative behaviors influencing the project completion. However, pursuing the prescribed relay race behavior rules in practice and sustaining them are experienced as a difficult part of the CCPM deployment, especially when different aspects in the organization, e.g. proper defining roles and responsibilities, impact the performance of executing- projects.

To define roles and responsibilities and requirements within each function, were also several documents provided and training workshops for the prospective CCPM practitioner's organized and internal procedures and agreements were made. The importance of the task management role and finding appropriate people for fulfillment of this function are found as a challenging success factor for CCPM deployment within the CCTV development department. Task managers are required to possess strong interpersonal skills for coaching of the task performers, and spreading the relay race mentality among the task performers. The experiences at the Bosch CCTV, however, showed that it is not easy to find qualified and talented people for this function.

5.2.3 Piloting CCPM at Bosch CCTV

At the time of this research, the utilization of CCPM multi-project approach at the CCTV development department was still in the experimental stage and it is not applied in practice. This happened while the launched pilot project was executing in a multi-project environment. Depending on volume and cycle time software projects are usually decomposed to the manageable number of sub-projects which use a common resource pool. The CCPM piloting project included four steps (increments). Each consisted of ten interdependent mini-projects with the common resource pool (task performers, software architects etc). The interdependency of these projects was defined by so-called integration points where continuation of a particular mini-project was dependent on reaching (completing to a defined level) one or more integration points of preceding (prioritized) mini-projects. Unfortunately later this project was cancelled because of delays as a consequence of high technological uncertainties and disruptions. The internal evaluations of this failure indicate that neglecting of dependencies between the mini-projects was one of the major reasons of delay of the pilot project. To cope with these dependencies and to decrease resource contention in multi-project approach, CCPM suggests the prioritizing and staggering (pipelining) the projects before starting the projects. However, this did not happen because each mini-project was scheduled and executed separately. Also, because of the high technical complexity and uncertainty there was no guarantee that with proper implementing and pursuing of CCPM rules this project could be successfully completed on time.

As mentioned in Section 3.2, in the multi-project approach of CCPM another objective is targeted: the enhancement of the organization's throughput, i.e. the number of delivered projects with the same resources of the organization. Implementing the Critical-Chain multi-project approach requires first the identification of the drum resource (the most loaded resource shared across projects) in the multi-project organization which limits the flow of projects (throughput). Each project is then scheduled sequentially across the drum resource in a manner to maximize the utilization of the drum resource. CCPM claims reducing the overloading of the drum resource between projects in a multi-project environment will increase the rate at which projects are completed (throughput). This requires the organization's support for prioritizing and pipelining of organization projects, and also needs more coordination effort than Critical-Chain single project approach. Hence, the piloting of CCPM at multi-project level requires also more preparations and coordination effort. Furthermore, there is no guarantee that through piloting the objective as throughout improvement to be achieved. Therefore, selecting and conducting a complex and challenging software project, which consisted of multiple interdependent mini-projects (multi-project environment), for piloting of the CCPM rules was not a well-considered decision for CCPM implementation at the Bosch CCTV division. Piloting of CCPM rules in the multi project level needs high preparation and coordination effort. The earlier attempts on the single project level can be helpful for gaining initial experiences and skills to accustom with CCPM rules when deciding to go through CCPM multi-project approach. However, early piloting at CCPM multi-project level without sufficient training and skills and lack of support for the prioritizing and staggering (pipelining) of mini-projects, and consequently insufficient attention on the interdependencies of mini-projects through integration points, made the CCPM piloting a risky adventure within the development division which led to a failure.

5.2.4 The encountered difficulties in the implementing CCPM pilot-project

In this section we give an overview of the perceived improvement areas for the CCPM implementation. These are those implementation aspects (section 5.2.2) which are not sufficiently attained at the Bosch CCTV development department and need to be further improved.

a) Obtaining task performers' endorsement; The existing guidelines and the provided CCPM references or documents by Realization were for a great part CCPM tool-related and were not specifically developed or sufficiently adjusted for software development projects. Also, the involved consultants did not possess a technical or project-related background or a good understanding of software development process to smooth out the CCPM application problems properly. As we described in Chapter 4, some of the Critical-chain rules are not completely applicable or compatible with the nature of projects of Bosch development department, e.g. coping with project risks and swapping of resources as CCPM supposes.

b) Shifting focus from individual tasks to project due date; experiences at the development department showed that this CCPM emphasis often is not adhered by the CCPM practitioners. Too much focus on the running-tasks (WIP tasks), and short term accomplishments, and less initiative for improving and recovery actions were evidences for veering this CCPM emphasis. This resulted increasingly in loss of quality of completed work, e.g. mistakes or rework and increasing dissatisfaction among task performers in the CCPM pilot project.

c) Clear defining and assigning role and responsibilities; despite the importance of task management role in the Critical-Chain deployment, the appointed persons to this function were not fully qualified or capable for this role. To meet the requirements of this function, it is required that task managers besides the task related (technical) knowledge and experience, also possess fully understanding of the Critical-Chain rules and concepts and strong interpersonal skills and. Also, there was still confusion over the exact responsibilities and contents for this function that resulted in not sufficient adhering to the CCPM rules and prescribed procedures. Also the supporting-CCPM package was not properly utilized because of lack of buy-in for the CCPM multi-project approach and proficiency in the multi-project features of the CCPM tool.

d) Changing performance-measurement attitude; the buffer status is not only used as a measuring tool for the pilot project progress but also it was used for the judgment for the performance of the task performers at the development department. Critical-Chain approach recognizes the likely delays of tasks as result of setting aggressive task durations and tight schedules and assumes that the task overruns can be captured through inserting-buffers in the strategic places of the project network. Also, it prescribes the relay race mentality to take the advantages of early finishes of tasks which contributes to the compensation of project buffer. The almost constant reporting of the red buffer status (going into the red zone of the fever chart) as a consequence of high tasks overrun because of complexity of tasks and high technological uncertainty, however, observed by the managers partly as a result of the insufficient performance of the project team. Hence, the red buffer status was experienced by the task performers as a warning for coming tough measures by the management, e.g. overwork. While there were usually different reasons for high buffer depletion rates or even too soon total depleting of project buffer, e.g. appearance of unforeseen new tasks and significant changes in the project scope.

e) Sustaining of implemented changes; despite the initial CCPM buy-in by the task performers, setting overly aggressive and unrealistic task duration estimates, and tight schedules resulted in gradual loss of CCPM support among the task performers. Constantly being confronted with delays and being in the critical red zone of the fever chart as a consequence of high project buffer deletion rate caused gradually dissatisfaction and sandbagging, especially when despite devoting sufficient effort on executing tasks in pilot project, the project buffer totally depleted even at the beginning period of project execution. This happened because the inserting buffers were not able to capture the high amount of technological uncertainties and to incorporate unforeseen new tasks as result of significant scope changes in the course of the pilot project.

f) Existing resistance to CCPM deployment and roll-out; resistance existed to the further expanding of the CCPM deployment for all projects of the CCTV development division, e.g. hardware development, and to the company-wide implementation of CCPM pipelining approach. The technical project personnel considered the technical complexity of developing-product as the main challenge in their mission, and were not concentrated in the issues as the project scheduling and monitoring. The CCPM however takes the time dimension into account and does not address the other two important project’s dimensions like budget and quality. Also, despite the success stories about the benefits of the CCPM rules and assumptions for improving the project scheduling and tracking, efficient and detailed procedures for applying this technique to a portfolio of projects, i.e. multi-project environment are still unclear. The evidences of its success are still almost exclusively anecdotal and based on single case studies and simplistic depictions. To date no large scale empirical research exists to prove its efficacy.

5.3 Summary and Conclusion

In this chapter we explored the important success factors in the CCPM implementation. These aspects are obtained through the experiences and findings reached by many CCPM implementation attempts within the different companies. Attaining the senior management support and involvement for the CCPM deployment, understanding of Critical Chain implications and concepts and changing measuring attitude, shifting focus from local toward global view, i.e. from individual tasks to whole project due date, clear determining role and responsibilities and assigning them to appropriate people, proper communication with both internal and external clients and suppliers of the CCPM organization, and sustaining the implemented changers are important implementation aspects for the CCPM implementation within an organization.

Also, piloting of CCPM rules and prescriptions is a useful way for practicing of the CCPM rules and concepts, exploring the required changes, and analyzing the outcomes to decide for the further implementing of CCPM system or rolling out to the rest of organization. Piloting of CCPM single approach is easy while in multi-project environment requires more coordination and preparation effort, because of required decisions and agreements for prioritizing and pipelining of multiple projects which sharing a common resource pool. Also, though piloting of CCPM multi-project approach, analysis of outcomes like throughput are not aimed to achieve, otherwise the whole projects of company or particular project portfolio should be added in the pipeline schedule and executed.

Study of the CCPM implementation at the Bosch CCTV showed that not all mentioned implementation aspects are sufficiently realized or maintained within the development department. The highly uncertain environment of product development projects and selecting a complex software project for piloting of CCPM system that were executed in a multi-project domain, i.e. including multiple mini-projects with a common resource pool, contributed also to this failure. Appearance of unforeseen new tasks in the schedule, high or even unrecoverable rate of buffer depletion because of high technological uncertainties and risks were major reason for too soon confronting the red buffer status for the CCPM pilot project.

Also, the critical-chain tasks in pilot project received the most focus which led to the loss of focus on the new critical and complex tasks which had an impact on the status of whole project. Furthermore the appointed people were not sufficiently capable for their roles. Also, there existed some confusion about the exact content of their functions, e.g. task management. Furthermore, the continual reporting of the project red buffer status or total (unrecoverable) depletion of project buffer, led to gradual loss of CCPM buy-in among the task performers, while the encountered problems were for a great part caused by the huge complexity and consequently created high uncertainties which could not be totally captured by the allocating buffers in the project network schedule. This is, however, the CCPM deficiency in the high uncertain environments. Therefore, exploring and recognizing of the CCPM deficiencies or abilities help to sustain the obtained endorsement of project team. Beside these difficulties we observe some positive aspects in the CCPM implementation within the department: The senior- management support and involvement for CCPM deployment at the development department and obtaining the project staff endorsement and understanding of the CCPM concepts and implications, and increasing consciousness about the required changes, e.g. negative behaviors which impacting the project due date through organizing training workshops and practicing of the CCPM rules by means of a pilot project were the positive attempts done by the CCTV development department. In the next chapter we address the most significant problems described in this chapter and provide some improvement suggestions.

6 Improvement proposals and recommendations

This chapter provides an answer to the fourth research question.

- What areas need improvement? What possibilities are there for improvement?

Referring to the encountered difficulties in the CCPM implementation at the Bosch CCTV Development (5.2.4), and the perceived deficiencies for deploying the CCPM rules and concepts in development projects (4.4.4), we formulated some improvement proposals. Because of the complexity of encountered problems we addressed the significant issues regarding the CCPM deployment and roll out within the CCTV development department. These are provided through reviewing CCPM literature and publications, experiences and lessons learned at the Bosch CCTV development division, and our findings in the course of this research.

6.1 Suggestions for improving the encountered problems at Bosch CCTV

Adjusting of tasks estimate and buffers size regarding the project risks and uncertainty; to cope with both common and specials causes of uncertainty or risks, and setting realistic task estimates and buffers, we recommend addressing the technological uncertainties and risks, and deploying risk management techniques. Repeating total buffer depletion (referred to 4.4.4.a) in pilot project occurred for a great part due to significant scope changes and appearance of new tasks in the planning as a consequences of the technical uncertainties that could not be captured by initial determined buffers that CCPM prescribes. CCPM however addresses only common causes of variation. Consequently, allocating buffers to the project schedule without addressing the technical uncertainties and complexities incorporated in the development projects will not provide sufficient protection. Therefore, to cope with the project technical risks, and to alleviate extreme buffer depletion rate as a consequence of risks, we recommend to also incorporating the risk management activities in the project planning and consider them in the buffer sizing calculations.

Re-scheduling CCPM project schedule in case of significant scope changes and total buffer depletion; in the literature, the frequent rescheduling of project schedule is discouraged because shifting focus of the project team. However, the re-scheduling in the high uncertain product development environment is not inevitable completely because the allocating buffer cannot absorb all kind of uncertainties existing in the development projects. As we experienced in the CCPM pilot project, the allocating-buffers were not able to absorb special causes of variation in the project, e.g. technological interruptions and risks, and unforeseen scope changes and addition of new tasks. Therefore we recommend in case of total or unrecoverable buffer depletion to evaluate the reasons for this and address them in the next scheduling attempts. Hence, the buffer status, should not be directly used for measuring the performance of the tasks performers (referred to 5.2.4.e). Re-scheduling is required only when significant changes in the project scope are happened or when the project buffer is totally depleted and is not recoverable anymore. As mentioned, incorporating the risk assessment and risking mitigation, avoidance or prevention activities, and considering these activities in sizing the project buffer decreases the need to frequently reschedule. Re-scheduling, however, provides an update-network schedule regarding the developments in the project scope and number of project tasks in the course of project execution. Further, it is required to early inform the project stakeholders and clients to about the new schedule and the new project due date and to reconsider the new situation.

Combining of technical dependent tasks regarding resources interchangeability problem; for the tasks with the internal hand-offs dependencies, i.e. the output of the predecessor tasks is the input to start the successor task by the same resource(s). In the literature is recommended that these tasks should be incorporated into one task in the project schedule (Austin and Peschke, 1999). Otherwise, when constructing a (resource constrained project CCPM) schedule, the resource dependent tasks are leveled or postponed to other places in the project network schedule where sufficient resources of the same type are available. This is not always appropriate because of leveling of these tasks to the later periods in the project network or performing them by other task performer takes more time because of the required preparations (referred to 4.4.4.f). For example, for each component X, Y or Z in Figure 4.4, the tasks like Specification, Design, and Coding are incorporated in one task (Figure 4.4-B). By incorporating of the tasks with internal hand-offs within a bigger task for each component, the activities for each component are executed by the same task performer. This provides more focus and less preparation time and prevents the task performers against multitasking. The disadvantage of this solution is in this way the tasks in the project schedule become larger and consequently the tracking of the executing items within the task scale become more difficult. Therefore, it is required to make a detailed list of the containing-activities and accomplishments when executing these tasks.

Explicitly defining and assigning CCPM roles, responsibilities and procedures; the CCPM concepts and rules are general and are not developed for a specific organization or project type. Depending on the organization's structure and the nature of projects, clear references or guidelines regarding the CCPM rules and necessary changes for applying them need to be provided. Further clear role definition, determining responsibilities, and assigning them to appropriate people are required. For example, CCPM suggests frequently (daily) updating of the completed, in progress and upcoming tasks. How data gathering and updating should take place, depends on the reached agreements and defined procedures within each organization. Referring to the dissatisfaction of the task performers about the time consuming monitoring procedures at the development department (referred to 4.4.4.d), we note that gathering of update-information and issues resolution are continues processes during the project execution and should be done in an efficient and sustainable way to achieve and maintain cooperation of the project participants. Setting of time-consuming procedures and meetings may cause loss of CCPM buy-in and stimulate the negative resources behaviors. Also to implement the relay-race mentality and maintaining it among the task performers and to remove negative resource behaviors, capable and qualified task managers with a good insight into human character are required. Further, the shortened task duration estimates in the aggressive project schedules and the sized buffers should create a sense of dedication for the task performers and be targeted to reach with the sufficient amount of efforts by the resources. Also, a good substitution should be sought, when a task manger needs to be replaced in case of absence or leaving.

Using standby supporting- resources and upgrading the resources usability and interchangeability; to cope with the resource substitution and interchanging difficulties (referred to 4.4.4.g), and supporting recover actions in order to deal with the likely addition of unforeseen new tasks, interruptions, and facilitating supporting-activities for the task performers during their task mission, we recommend not to plan all resources into the CCPM schedule. These task performers can operate then as supporting-resources for facilitating and removing possible interruptions in executing-tasks, or when they are in the red buffer status, or in the case of absence or sickness of task performers working on them. Meanwhile these supporting-resources can be trained or involved in the specifications and requirements of the identified critical tasks in addition to the task performers working on these tasks, to support task performers if their tasks have the red buffer status. Also, to cope with the non-routine nature of software development tasks and consequently the difficulty in the resource transfer or swapping, we propose also to inform and involve some of the assigning-task performers in the content and requirements of the identified critical tasks when they are temporarily idle or have free time. This facilitates the resource interchangeability or transferability from the non-critical tasks to the executing-tasks with the red buffer status, if it is required. In the software development projects the tasks are often not routine and consequently when a task performer is assigned to a new task, it is often required to know what the previous task performers have already done.

6.2 Proposals for the further CCPM-implementation and roll-out

Deploying the CCPM Pipelining approach separately for each projects portfolio; The development projects in each portfolio (Camera or DVR-digital video recorder) at Bosch CCTV Eindhoven are mostly interdependent and share a common team of task performers for executing- projects. Depending of scope of a project, this might be divided into number of manageable smaller sub-projects which are then also technically interdependent, i.e. reaching defined level of one or more sub-projects provides the input for starting/completing the next one. To deal with these interdependencies and to decrease overloading of resources, we propose to deploy the CCPM multi-project (pipelining) approach separately for the interdependent group of projects with technical and resource dependencies. Although CCPM prescribes the prioritization and staggering (pipelining) of the interdependent projects which used a shared resource pool, within the organization, to our view this is not applicable for the whole projects of the Bosch CCTV development division. Because of the specific specialization and expertise of projects, these are clustered in different portfolios, and each group possesses its own team of professionals. Also the projects within one project group are independent of the other. However, the resources are occasionally swapped over one group to another. The experiences at the department showed that most constraining (drum) resource is not the same between the different project portfolios i.e. Camera or DVR. Therefore, we propose to pipeline separately the interdependent projects using common resources team or having technical dependencies. Also, in case of identifying several constraining resources between each group of interdependent projects, the most bottleneck resource should be chosen.

Also, before deploying the multi-project approach, it should be considered what the Bosch CCTV development division primarily aims to achieve. As referred, the objective of CCPM multi-project (pipelining) approach is to increase the organization's throughput or the number of completed projects with the same organizations resources. CCPM claims that starting projects as soon as possible without regarding the availability of resources or organization capacity, increases the resource conflicts between the projects and obstructs the progress of each individual project in the multi-project domain. Therefore, to implement the CCPM pipeline approach, it is first required to prioritize all of an organization's projects (depending on organization's goals and market situation) and the stagger them across the most bottleneck resource of organization. As mentioned above, however, putting all organization's projects in a pipeline schedule is not always required or applicable in practice. This is not useful when an organization has separate project groups each with its own project specialization, resource team, and budget. Furthermore, consolidating of the development projects in a pipeline (drum) schedule requires integral view and agreements for the prioritizing or sequencing of organization projects. It requires also the subordination of managing the individual projects to the needs and goals of the entire organization. However, the projects priority is for a great part depending on the developments on the market and may drastic change during the execution of the pipeline schedule. Hence, it is then required to reflect on these changes in the pipeline schedule again and to reach new agreements within the department for re-pipelining, i.e. prioritizing and staggering of projects. This all discourages the deployment of CCPM pipelining for all projects of development department. Hence, we finally recommend deploying and rolling-out the CCPM pipeline approach separately for each portfolio or group of interdependent projects with impacting resource and precedence interdependencies.

7 Conclusions and suggestions for further research

This chapter summarizes the conclusions drawn in this research. This thesis aimed to survey the deployment of Critical Chain methodology for the planning and managing of the development projects, to explore important CCPM implementation aspects, and surveying how are these addressed at the Bosch CCTV development department, and finally to formulate some improvement proposals for the encountered problems.

Chapter 2 elaborated on the critical chain scheduling method and compared it with the traditional project scheduling. In this chapter we addressed the first research question about the major differences the Critical Chain Scheduling (CCS) versus the Traditional Project Scheduling (TPS). Coping with the existing uncertainty in the project schedule through setting aggressive schedules, by removing safety time from the individual tasks and aggregating them in time buffers, which are inserted to strategic places in the project network schedule to absorb the existing variations in the project duration, is the major difference between the Critical Chain scheduling method and the traditional scheduling methods. Critical Chain methodology also prescribes guidelines to deal with the negative human-resource behaviors which are not explicitly addressed in the traditional project scheduling. Also, we reviewed several critiques to the Critical-Chain approach. The significant one is that the Critical Chain however did not provide a solid basis for cutting-time of tasks duration and sizing of inserting-buffers which has a direct impact on the project duration.

CCPM also suggests a scheduling approach for the multi-project environment. Chapter 3 elaborates the CCPM prescription for the environment wherein multiple projects share a common resources pool within the organization. The CCPM suggestion for pipelining of projects, however, requires reaching many agreements for the identified drum resource, prioritizing of projects and staggering of projects across the drum resource. Also, in practice it is often difficult to identify exactly the single heavily loaded (drum) resource within the multi-project organization because of each of the professional teams or project portfolios had its own (drum) resource bottleneck. Further, because of changing-priorities of projects during the project execution and the developments on the market, e.g. in the product development environment, repeatedly re-pipelining (re-prioritizing and re-staggering) of competing projects and high coordination efforts are required. Furthermore, the suggested approach is introduced in a simplistic form through identical (resource-dependent) projects that in practice is certainly not the case. Hence, there is no warranty that the prioritizing and staggering of projects across the drum resource without regard to the number of tasks, their interdependencies and the different tasks each task demand will increase the throughput as it supposed.

In Chapter 4, we addressed the second research question involving the differences between the Critical Chain rules and concepts as supposed in the theory and the Critical Chain application at Bosch CCTV. The experiences at the development division indicate that Critical-Chain planning gives more visibility or transparency in the projects status as it is asserted. The findings of the CCPM users at the development department endorse the buffer-management feature of the CCPM approach as an effective advantage. Through monitoring the buffer status CCPM provides an early warning tracking system and helps the management to take measures by anticipating delays. We perceived, however, that because of existing high uncertainty and complex nature of the (software) development projects, some CCPM rules are not be sufficiently compatible or applicable in practice. For example, because of the high technological uncertainty, i.e. significant project scope changes or appearance of unforeseen new tasks, the allocated project buffer did not often provide sufficient protection for the original project due date. Hence, the frequent re-scheduling of the project network was inevitable. CCPM, however, does not address the special causes of variations and risks. Also, for recovery attempts, CCPM suggests transferring resources from non-critical (chain) tasks to the critical (chain) tasks with the red buffer status to recover or protect project due dates. This is, however, in the development projects not always feasible because the tasks in these projects usually contain non-routine activities. Hence, the new task performers need to acquire required knowledge about the executing tasks which enables them to complete their task.

To address the third research question, in Chapter 5, we provided an overview of the explored determining factors for the successful CCPM implementation and investigated how far these are achieved at the Bosch CCTV development division. Without regard to the outcomes, we discussed the usefulness of piloting for facilitating and analyzing the required changes for implementing CCPM. Piloting of CCPM single-project approach is easy while in multi-project environment requires wide coordination effort, because of required decisions and agreements for prioritizing and pipelining of multiple projects which sharing a common resource pool or technical interdependencies. Also, through piloting of CCPM multi-project approach, analysis of some outcomes like throughput are not likely to be achieved, because it involves mostly limited projects of the organization and is not a company wide deployment as we also perceived at the Bosch CCTV development department. Study of the CCPM implementation at Bosch CCTV showed that not the all critical aspects are sufficiently addressed within the development department. Also, the complex characteristics of developing projects and choosing a complex project with high technological uncertainty as a pilot project contributed to these failures. The continual red project buffer status or total (unrecoverable) depletion of project buffer reduced the CCPM support among the task performers, while the due date overruns were for a great part due to the complexity and technical risks which could not be totally captured by the CCPM buffers. Also resistance still existed to the CCPM methodology and company wide pipelining of projects. The reasoning is that despite the success stories about CCPM, efficient and detailed procedures for applying this approach to a portfolio of projects (multi-project environment) are still not provided.

Chapter 6 addressed the fourth research question and provided some recommendations and improvement proposals for the significant problems for the CCPM deployment and implementation described in Chapter 4 and 5.

7.1 Areas for the further research

Bosch CCTV Eindhoven looks at the CCPM deployment as a passed no-return point in its project's organization. The CCPM single approach is meanwhile deployed throughout the development division. The implementation experiences and outcomes from CCPM application (through pilot project) were evaluated, and efforts for improving its application and exploring possibilities for improvements are encouraged, that was the motivation behind this graduation study. The methods used in this study were interviews, a literature review, and experiences or findings to date at Bosch CCTV. The results are an overview of the perceived deficiencies in the CCPM application, the encountered difficulties in its implementation, and also provide different proposals and recommendations for improvement.

Regarding the outcomes of this study the following suggestions are made for further research:

Despite the perceived deficiencies and weaknesses for the CCPM implementation and deployment at the Bosch CCTV development department, we evaluated the Critical-Chain as a simple but effective approach for the scheduling and monitoring of the development projects if the projects risks are sufficiently addressed. Compared with the traditional project scheduling, CCPM is more predictable in terms of promised due date to both internal and external customers of a project in case of sufficient buffer protection. For the product development projects, meeting of due dates is one of success factors of introducing of a new product on the market. We observed that the sizing-buffers, however, do not provide sufficient protection against the specials causes of variation and inherent technological risks in the development projects. Therefore, in addition to CCPM the risk analysis and risk mitigation techniques should be applied to address the development project risks. CCPM enables the project managers to monitor the impact of uncertainty by means of buffer management, which helps them to operate proactively and be prepared for likely recovery actions in earlier stages of project execution to protect the promised project due date. Hence, to increase the CCPM protection and reliability in the high uncertainty domain of product development projects, integrating the risk management techniques with the critical chain methodology for avoiding or mitigating of risks is needed. As an extension to this research, further investigating this subject is suggested. Furthermore, the evidences of CCPM success are still almost exclusively anecdotal and demonstrated by simplistic depictions that are based on single case studies, and are mostly about its application in the manufacturing environment or the single-project implementations. To our knowledge, to date no large scale empirical research on the Critical-Chain multi-project approach exists to prove its effectiveness. This forms another obvious area for further research.

List of References

[1] Austin Kathleen (Kathy) M., and Peschke Richard E. (1999), The Goldratt Institute, How to Build a TOC/Critical Chain Project Network, APICS -The Educational Society for Resource Management

[2] Blazewicz, J., Lenstra, J.K., Kan, A H.G. Rinnooy (1983). Scheduling subject to resource constraints : Classification and Complexity. Discrete Applied Mathematics, 5, pp. 11-24.

[3] Cerveny, Janice F. (2005), Theory of Constraint International Certification Organization (TOCICO) Annual conference presentations, (http://www.tocico.org/)

[4] Devine, George (2004), Critical Chain Project Management: Controversy and Benefits, International Journal of Applied Management and technology, 2(1), 35-47.

[5] Goldratt, E. M. (1997), Critical Chain, , The North River Press, Great Barrington

[6] Goldratt, E.M. and Cox J. (1984), The Goal - A Process of Ongoing Improvement, North River Press, Croton-on-Hudson

[7] Gupta, S. (2005), Theory of Constraint International Certification Organization (TOCICO) Annual conference presentations, (http://www.tocico.org/)

[8] Hadar Ziv, Debra J. Richardson, and Ren'e Klosch, (1996), The uncertainty principle in software engineering, Technical Report UCI-TR-96-33, University of California

[9] Herroelen W. , Leus R, (2005b), Identification and illumination of popular misconceptions about project scheduling and time buffering in a resource-constrained environment, Journal of the Operational Research Society, vol. 56, no. 1 (Jan.), pp. 102 - 109.

[10] Herroelen, W. and Leus, R (2001), On the merits and pitfalls of critical chain scheduling Journal of Operations Management, (19) 5, 559-577

[11] Herroelen, W. and Leus, R., (2005a), Project scheduling under uncertainty—survey and research potentials, European Journal of Operational Research 165, 289–306

[12] Herroelen, W., (2006). Project scheduling - Theory and practice, Production and Operations Management, vol. 14, no. 4, pp. 413 - 432.

[13] Herroelen, W., Leus, R. (2004), Robust and Reactive Project Scheduling: A Review and Classification of Procedures, International Journal of Production Research, 42 (8), 1599-1620

[14] Herroelen, W., Leus, R., and Demeulemeester, E. (2002). Critical chain project scheduling: Do not oversimplify, Project Management Journal, 33 (4), 48-60

[15] Kendall Gerald I., Pitagorsky G. and Hulett D. (2001), Integrating Critical Chain and the PMBOK® Guide, International Institute for Learning, Inc.

[16] Kerzner, H. (2003), Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Wiley; 8 edition

[17] Leach Lawrence P. (2005), Critical Chain Project Management, 2nd Edition Artech House, Inc. Norwood, MA, hardcover

[18] Leach, L. P. (1999). Critical Chain Project Management improves project performance, Project Management Journal, (30) 2, 39-51.

[19] Lechler, T., Ronen B. and Stohr Edward A. (2005a), Critical Chain: A New Project Management Paradigm or Old Wine in New Bottles?, Engineering Management Journal, Vol. 17 Issue 4, p45-58, 14p

[20] Lechler, T., Ronen B. and Stohr Edward A. (2005b), Critical Chain Project Management: High Potential - Hard to Achieve, Stevens Institute of Technology.

[21] Lee B. and Miller J. (2004), Multi-Project Management in Software Engineering Using Simulation Modelling Using Simulation Modelling, Software Quality Journal, 12, 59–82

[22] Newbold, R. C. (1988), Project management in the fast lane: Applying the theory of constraints. APICS series on Constraints Management. New York: St Lucie Press.

[23] Özdamar, L and Alanya, E; (2001), Uncertainty Modelling in Software Development Projects (With Case Study), Annals of Operations Research, Volume 102, Numbers 1-4, pp. 157-178(22)

[24] Patrick, S.F. (1998), Program Management- Turning many projects into few priorities with TOC, Project Management Institute symposium, Philadelphia, Penn.

[25] Patrick, S.F. (1999), Critical chain scheduling and buffer management - getting out from between Parkinson's rock and Murphy's hard place, PM Network 13: 57-62.

[26] Patrick, S.F. (2001), Critical Chain and Risk Management - Protecting Project Value from Uncertainty, Focused Performance, Hillsborough. USA.

[27] Pritsker, A., Allan, B., Watters, L.J., Wolfe, P.M., (1969), Multiproject scheduling, A zero-one programming approach. Management Science, 16(1): pp. 93-108.

[28] Product Development Institute (1999) Tutorial: Goldratt's critical chain method, a one-project solution. (http://www.pdinstitute.com)

[29] Rand, G.K., 2000. Critical chain: Theory of constraints applied to project management. International Journal of Project Management18 (3), 173–177.

[30] Raz, T., Barnes, R. & Dvir, D. (2003). A Critical Look at Critical Chain Management. Project Management Journal, 24-32, December.

[31] Realization Technologies, Inc. (2003), Concerto TM software. (http://www.realization.com)

[32] Simpson, W.P. and Lynch, W. (1999), Critical Success Factors in Critical Chain Project management, Proceedings of the 30th Annual Project Management Institute Seminars and Symposium.

[33] Standish Group (1999), report: Chaos: A Recipe for Success, Standish Group International, http://www.standishgroup.com/sample_research/index.php.

[34] Steyn, H. (2002). Project management applications of the theory of constraints beyond critical chain scheduling. International Journal of Project Management 20: 75-80.

[35] Tukel Oya I., Rom Walter O., and Duni Eksioglu S. (2006), An investigation of buffer sizing techniques in critical chain scheduling, European Journal of Operational Research 172 401–416

[36] Zultner, Richard E. (1998), Critical Chain, Doing development faster with quality, Sixteenth Annual Pacific Northwest Software Quality Conference